DataFlow ExaScale SuperComputing: Revisiting the Paradigm and the Algorithms

V. Milutinovic, G. Rakocevic, S. Stojanovic, and Z. Sustran

University of Belgrade

Oskar Mencer

Imperial College, London

Oliver Pell

Maxeler Technologies, London and Palo Alto

Michael Flynn

Stanford University, Palo Alto, USA

Nenad Korolija

A 2013 Summer Intern @ Google, Mountain View, USA

Essence of the Paradigm:

For Big Data algorithms
and for the same hardware price as before,
achieving:

a) speed-up, 20-200

b) monthly electricity bills, reduced 20 times

c) size, 20 times smaller

The major issues of engineering are: design cost and design complexity.
Remember, economy has its own rules: production count and market demand!

Conditions that Must Hold:

LOOP ORIENTED:

1a. Over 95% of run time in loops [loops to almost zero?]

1b. Over 3x data reusability (e.g., $x+x^2+x^3+\ldots$) [how close to zero?]

APPLICATION ORIENTED:

2a. BigData [for data streaming, not for data control]

2b. MicroLatency [military and defense]

ENVIRONMENT ORIENTED:

3a. MAXJ [new programming model]

3b. WORM [prog effort+comp time]

Use a Lamborghini, not a Ferrari, to drive over a plowed field

European Dimension of the Paradigm:

Absolutely all results achieved in Europe:

a) All hardware produced in Europe,
specifically UK

b) All system software generated in Europe,
EU and WB

Stanford + Purdue

ControlFlow vs. DataFlow

ControlFlow (MultiFlow and ManyFlow):

- Top500 ranks using Linpack
  (Japanese K, IBM Sequoya, Cray Titan, ...)

DataFlow:

- Coarse Grain (HEP) vs. Fine Grain (Maxeler)

The history starts in 1960's!
The enabler technology did not exist before the year 2000!

Is the Top 10 Supercomputer List Relevant?

World's Top 10 Supercomputers

Position	Machine	Architecture
1	***Tianhe-2***	***x86 + Xeon Phi***
2	***Titan***	***x86 + GPU***
3	*Sequoia*	*BlueGene/Q*
4	*K Computer*	*SPARC*
5	*Mira*	*BlueGene/Q*
6	***Piz Daint***	***x86 + GPU***
7	***Stampede***	***x86 + Xeon Phi***
8	*JUQUEEN*	*BlueGene/Q*
9	*Vulcan*	*BlueGene/Q*
10	*SuperMUC*	*x86*

What would be their performance for zetadata (21), yotadata (24), brontodata (27)?

The 5V of BigData: Volume, Variety, Velocity, Variability, and Veracity (Hurson)

Essence of the DataFlow Approach!

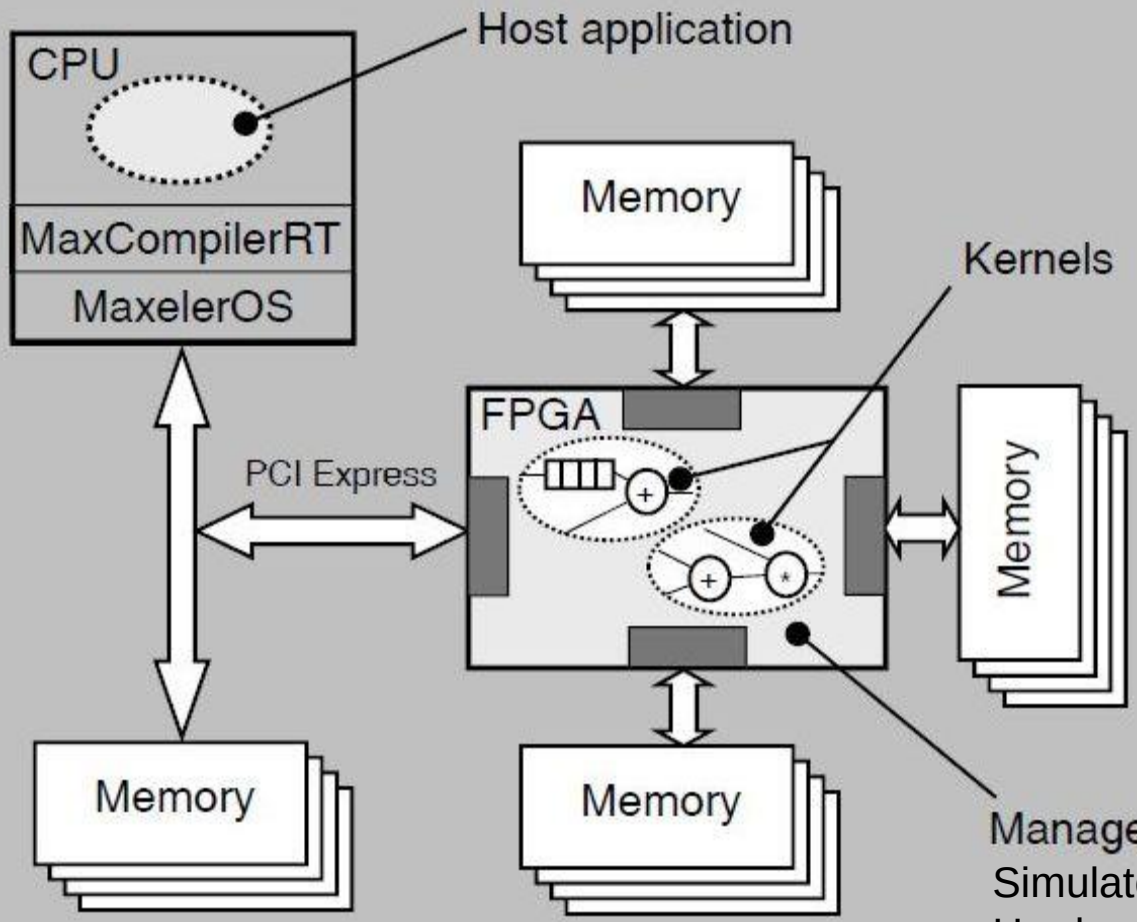
Compiling below the machine code level brings speedups;
also a smaller power, size, and **cost**.

The price to pay:
The machine is more difficult to program.

Consequently:
Ideal for WORM applications :)

Examples using Maxeler:
GeoPhysics (20-200), Banking (200-2000),
M&C (New York City), Datamining (Google), ...

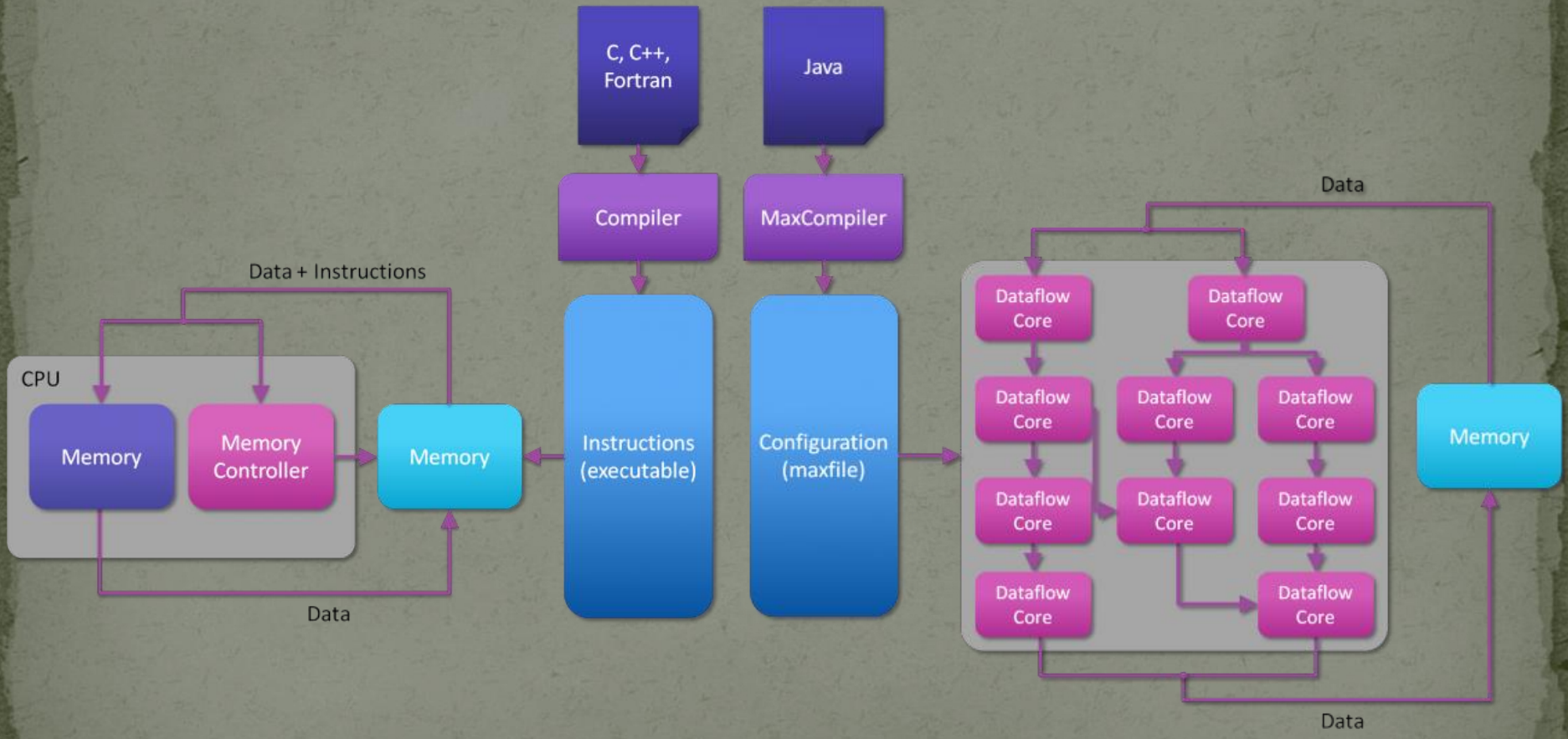
Generic Acceleration Architecture



MAXELER Technologies
MAXIMUM PERFORMANCE COMPUTING

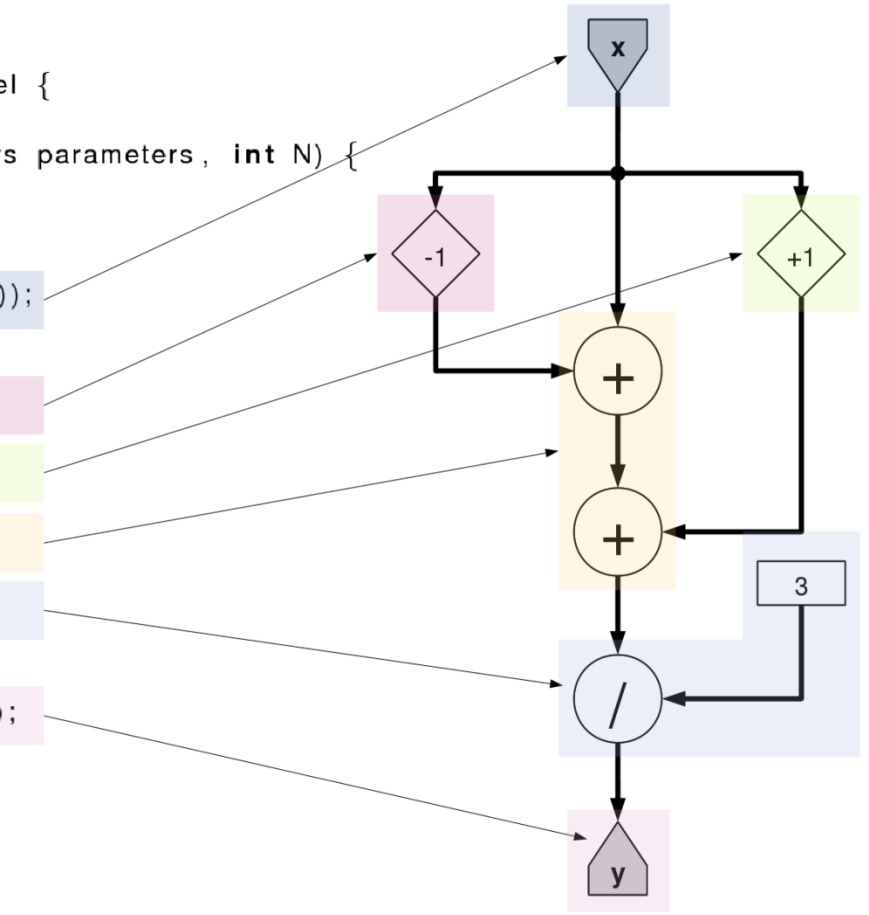
2n+3

ControlFlow vs. DataFlow



DataFlow Programming [1*C +(2n+3)*Java]

```
 7  public class MovingAverageKernel extends Kernel {
 8
 9      public MovingAverageKernel(KernelParameters parameters, int N) {
10          super(parameters);
11
12          // Input
13          HWVar x = io.input("x", hwFloat(8, 24));
14
15          // Data
16          HWVar prev = stream.offset(x, -1);
17
18          HWVar next = stream.offset(x, 1);
19
20          HWVar sum = prev+x+next;
21
22          HWVar result = sum/3;
23
24          // Output
25          io.output("y", result, hwFloat(8, 24));
26      }
27  }
```



Why Java? Minimal Kolmogorov Complexity, etc…

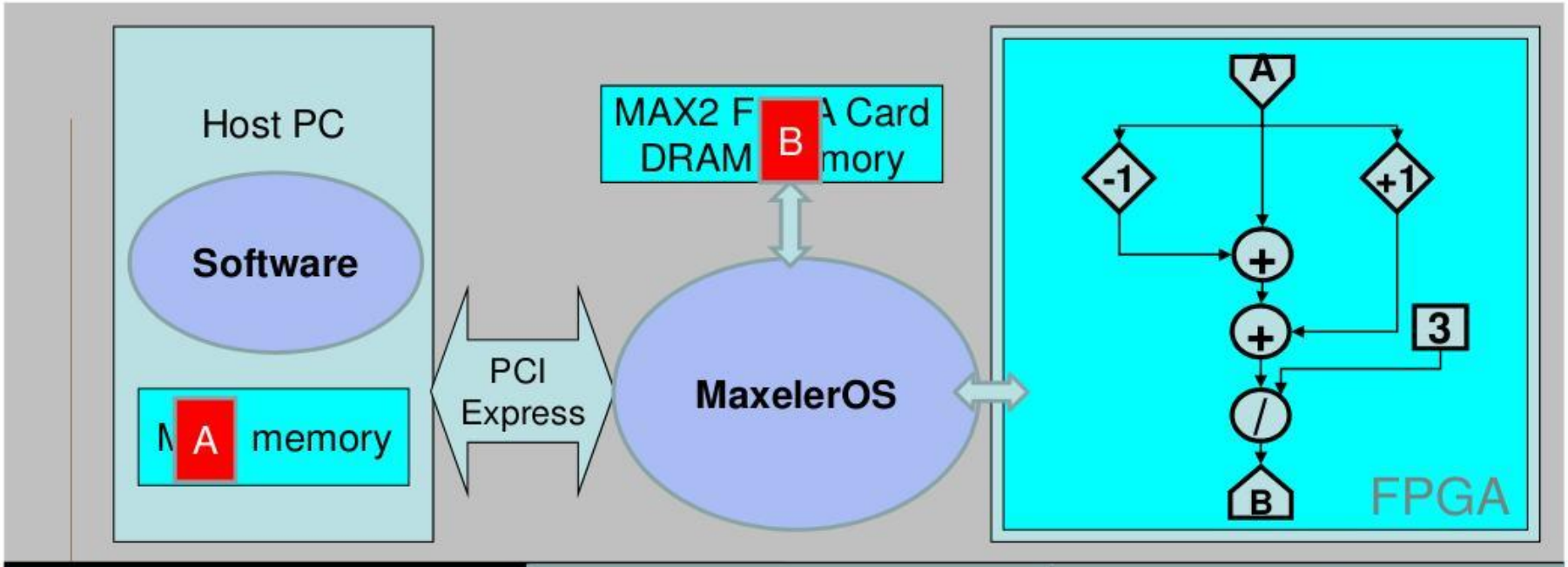
Kolmogorov Complexity, 1965

Definition (Kolmogorov):
"If a description of *string* *s*, *d*(*s*),
is of minimal length, [...]
it is called a **minimal description** of *s*.
Then the length of *d*(*s*), [...] is the **Kolmogorov complexity** of *s*,
written *K*(*s*), where *K(s)* = |*d(s)*|"

Of course K(s) depends heavily on the ***Language L***
used to describe actions in K
(e.g., Java, Esperanto, an Executable file, etc).

Kolmogorov Complexity => minimal energy to move Data

Kolmogorov, A.N. (1965). "Three Approaches to the Quantitative Definition of Information". *Problems Inform. Transmission* **1** (1): 1–7.



Software

C

```
device = max_open_device(
         maxfile, "/dev/max0");
float A[SIZE];
...
stream_data(device, A);

for (int i=0; i<SIZE; ++i) {
  B[i] = ( A[i-1] + A[i] + A[i+1] )/3;
}

...
```

Manager

Java

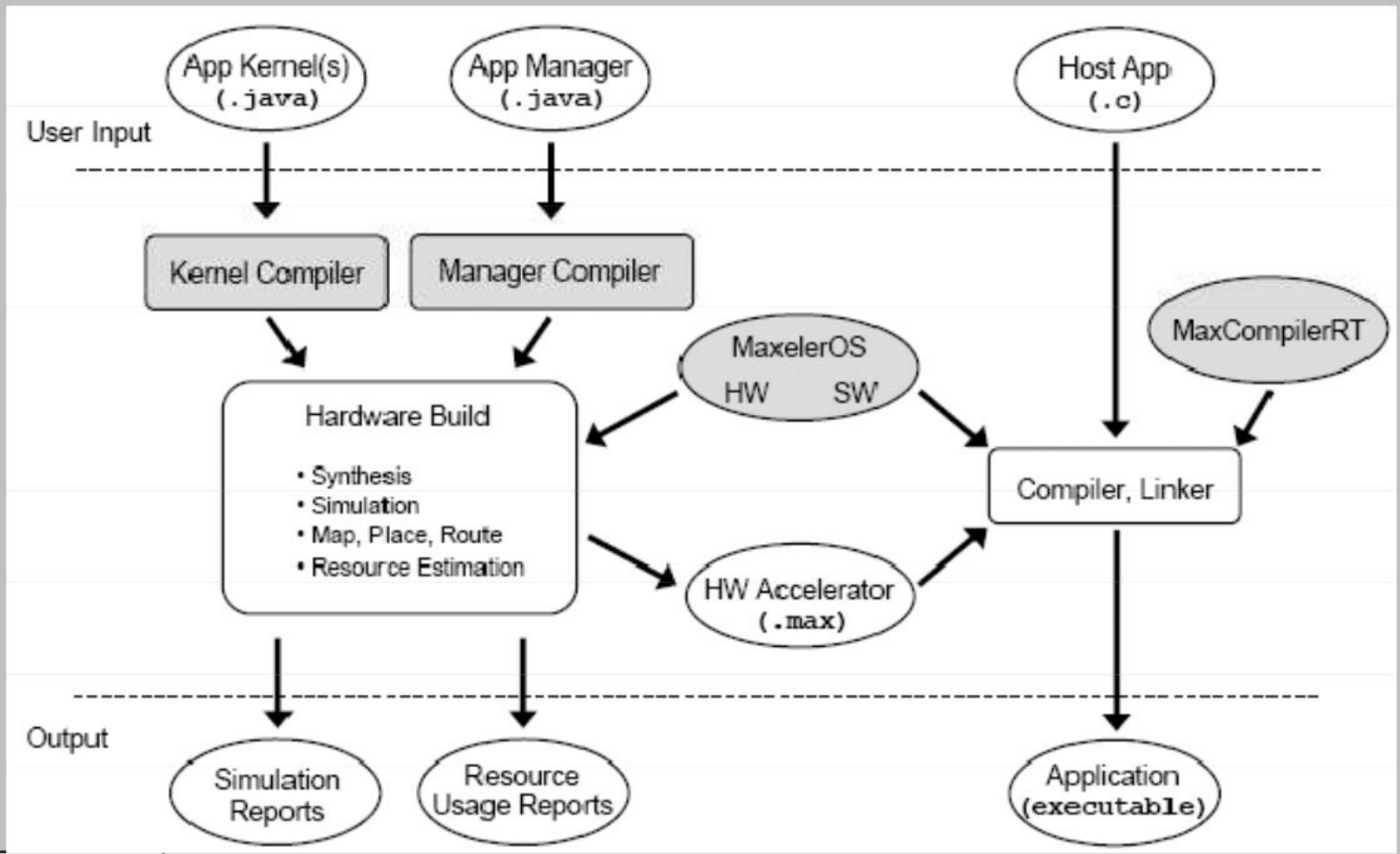
```
Manager m = new
    Manager("Loop", MAX2);

m.kernel(mav_kernel,
      link("A", PCIE),
      link("B", DRAM(LINEAR));

m.build();
```

Kernel

MaxJava

```
class mav_kernel
            extends kernel{

  input ("A",hwFloat (12 ,52) ) ;
  output ("B",hwFloat (12 ,52) ) ;

  A_prev=streamOffset(-1,A);
  A_next=streamOffset(1,A);

  B = (A_prev+A+A_next) / 3 ;
}
```

MaxCompiler

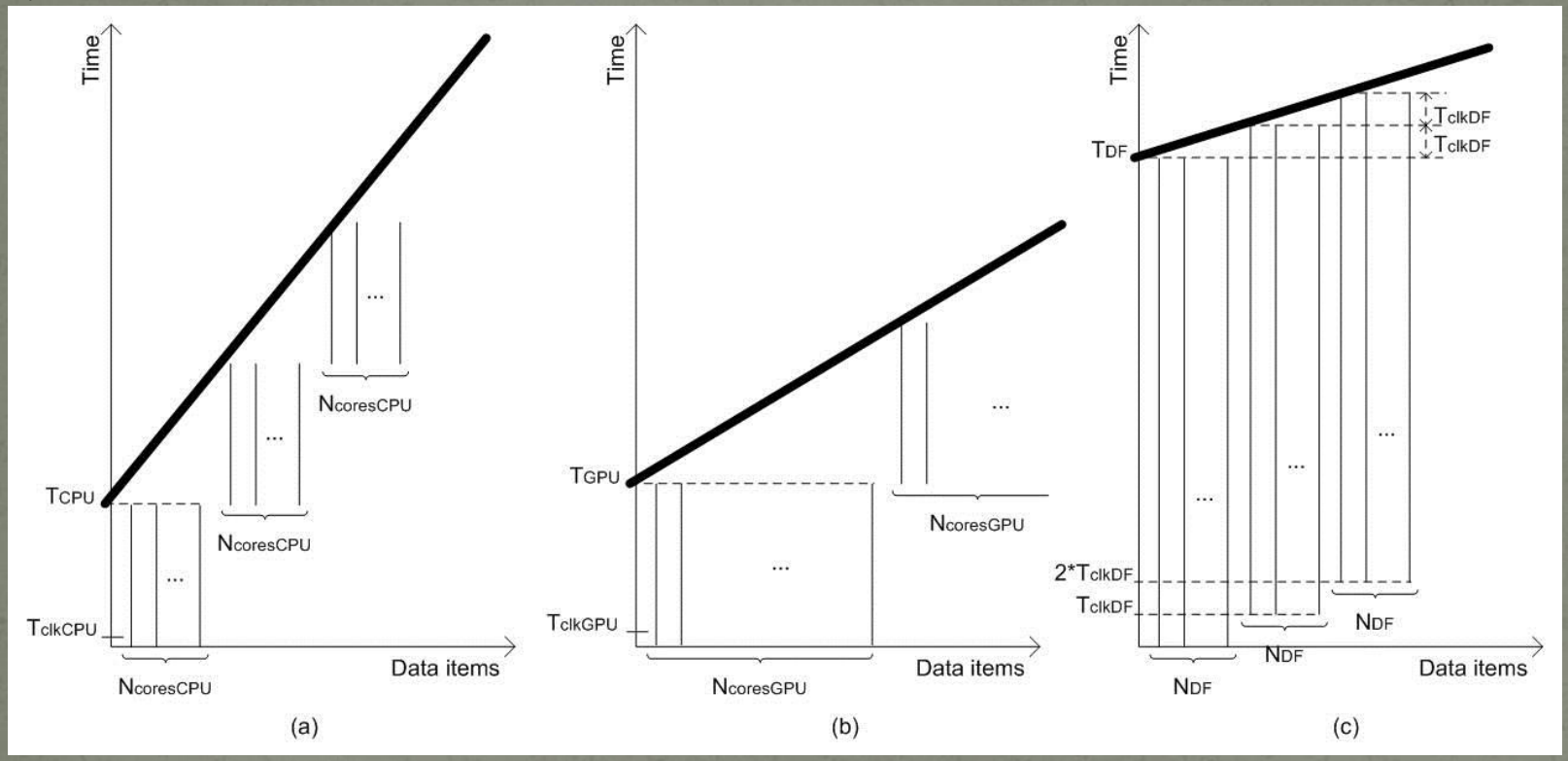


Essence: Paper by Sasa Stojanovic

tCPU =
*N * NOPS * CCPU*TclkCPU /NcoresCPU*

tGPU =
*N * NOPS * CGPU*TclkGPU / NcoresGPU*

*tDF = NOPS * CDF * TclkDF + (N – 1) * TclkDF / NDF*



Assumptions:

1. Software includes enough parallelism to keep all cores busy
2. The only limiting factor is the number of cores.

MultiCore

DualCore?

Which way are the horses going?

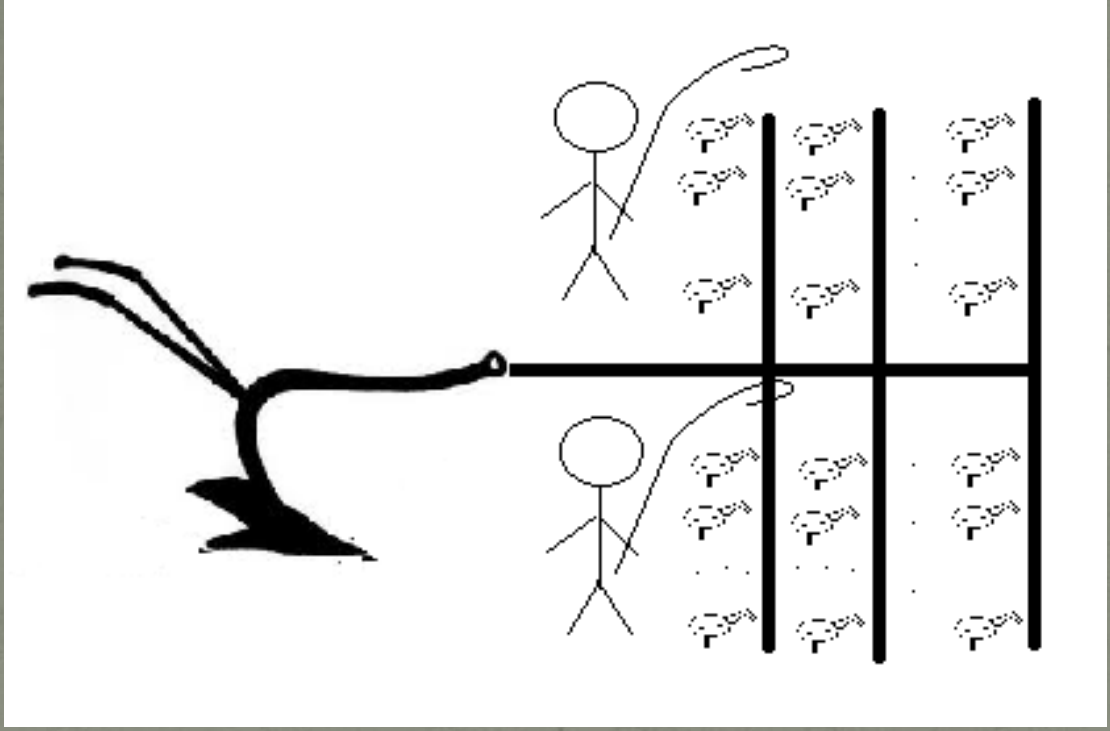
ManyCore

- Is it possible
  to use 2000 chicken instead of two horses?

?
==

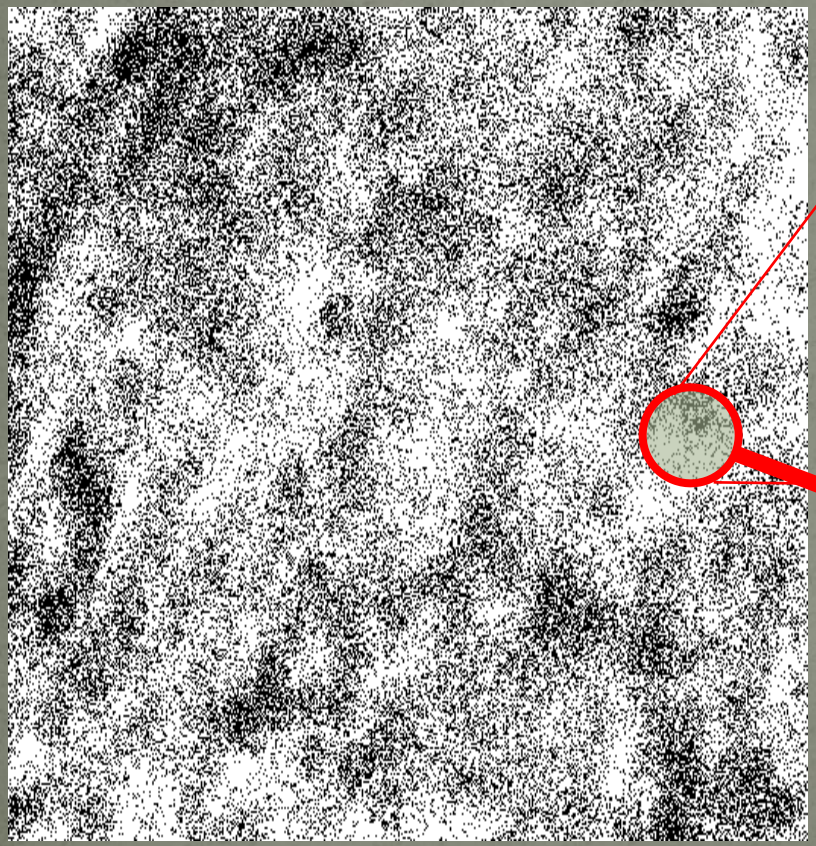
- What is better?

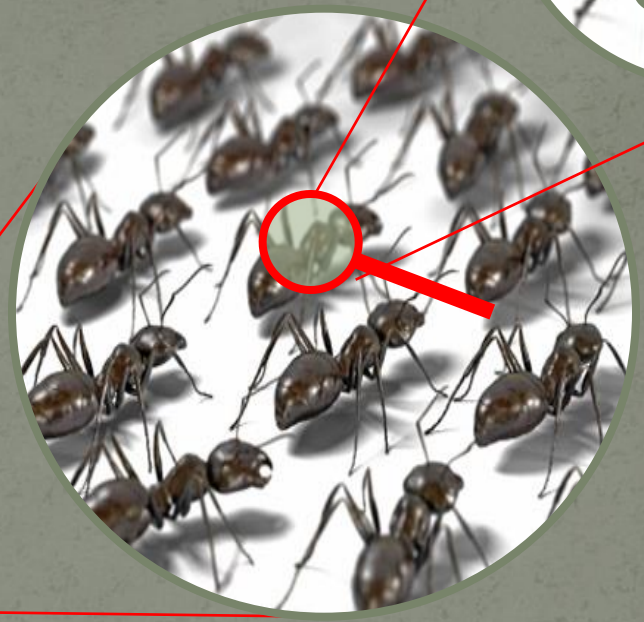
ManyCore



2 x 1000 chickens (CUDA and rCUDA)

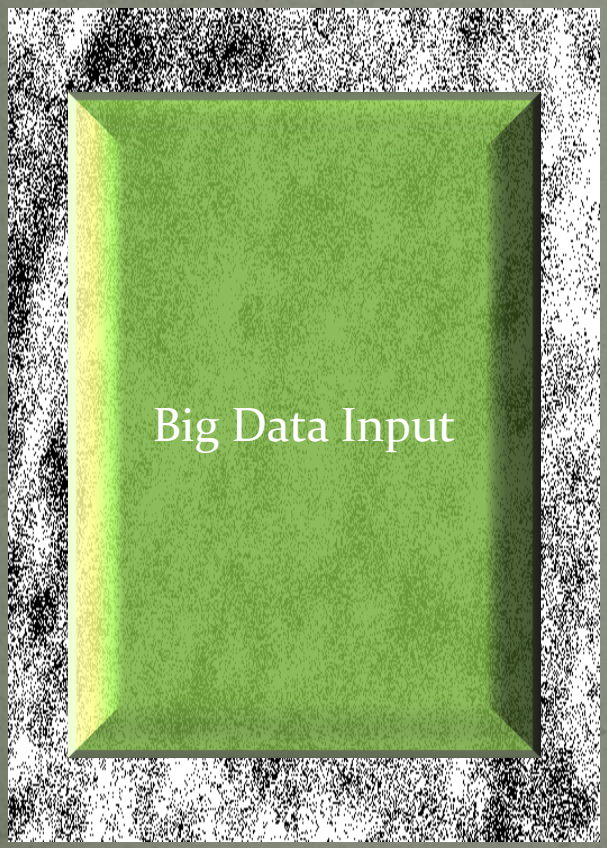
DataFlow

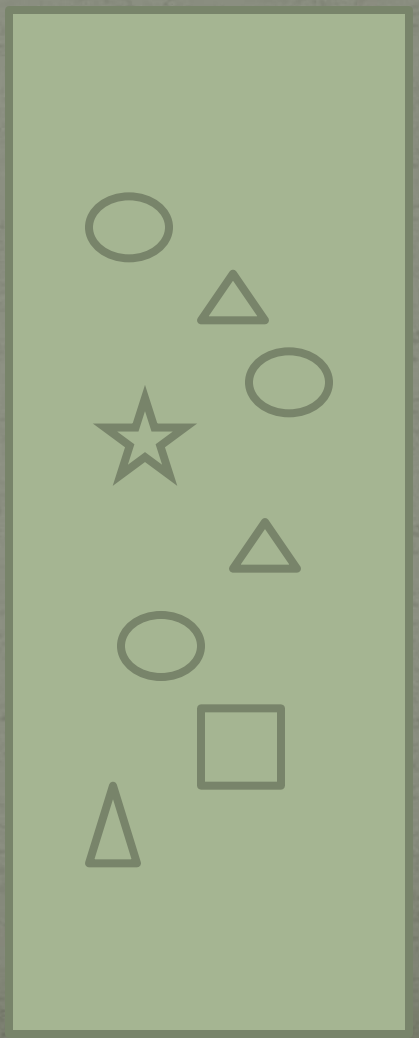




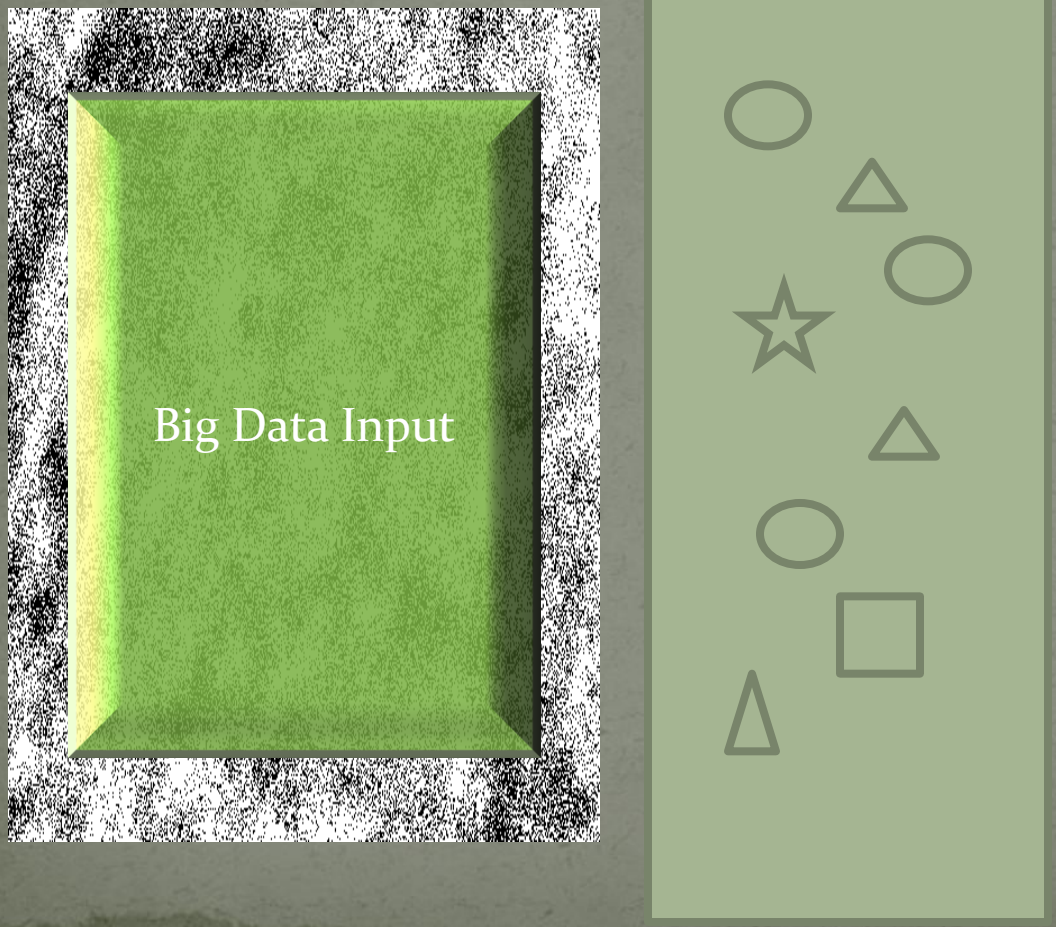
How about 2 000 000 ants?

DataFlow

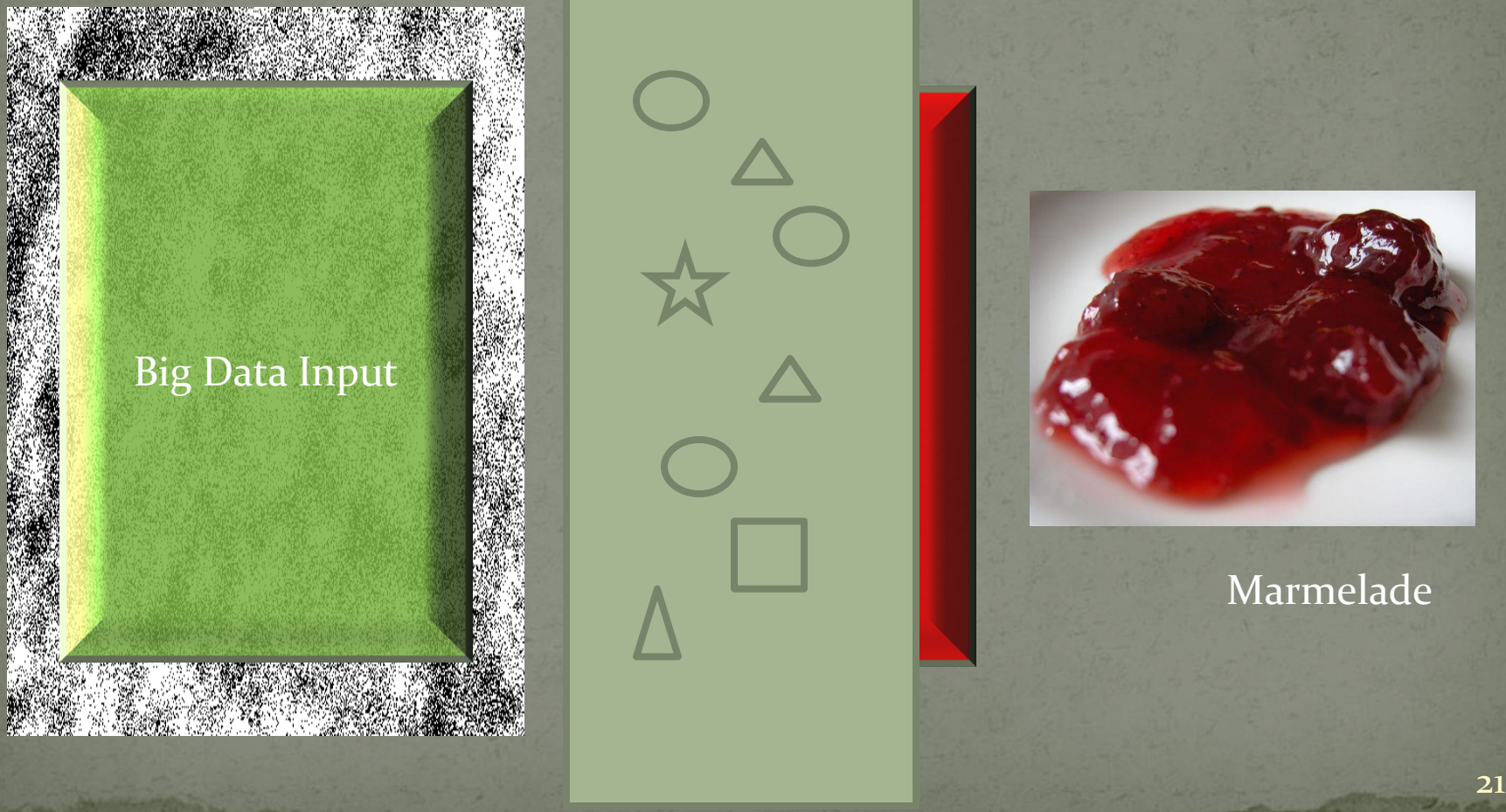




DataFlow



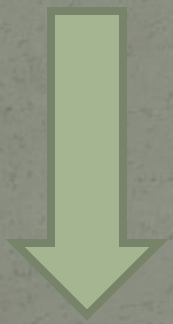
DataFlow



Why is DataFlow so Much Faster?

- Factor: 20 to 200

MultiCore/ManyCore



Machine Level Code

DataFlow

Gate Transfer Level

Why are Electricity Bills so Small?

- Factor: 20

MultiCore/ManyCore

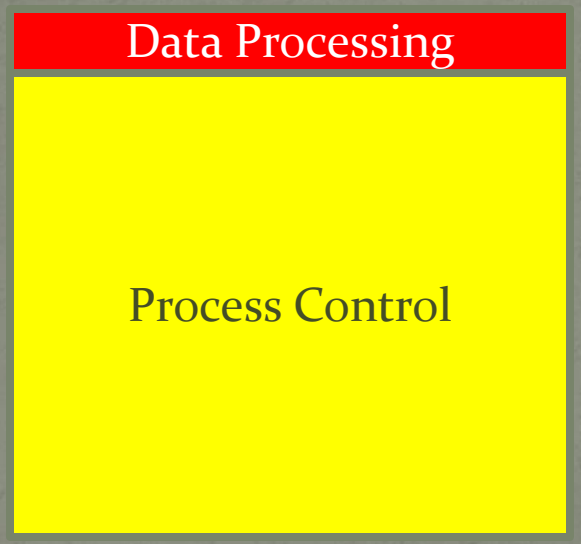
DataFlow

$$P = kfU^2$$

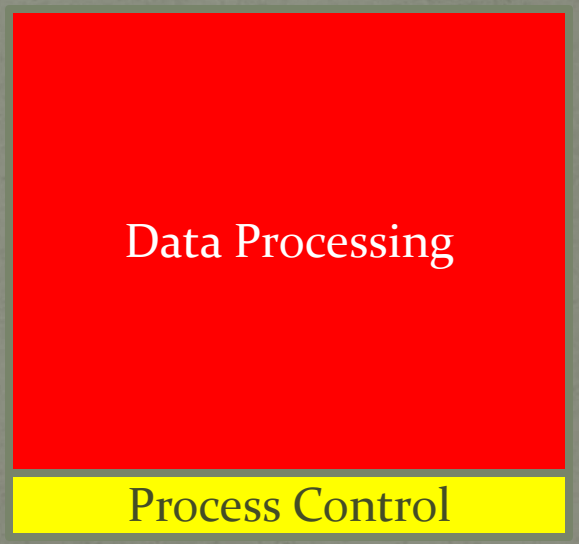
Why is the Cubic Foot so Small?

- Factor: 20

MultiCore/ManyCore



DataFlow



Required Programming Effort?

- MultiCore:
  - Explain what to do, to the driver
  - Caches, instruction buffers, and predictors needed
- ManyCore:
  - Explain what to do, to many sub-drivers
  - Reduced caches and instruction buffers needed
- DataFlow:
  - Make a field of processing gates: 1C+2nJava+3Java
  - No caches
- 300 students/year: BGD, FRI, BCN, UNISI, ETH, EPFL, …

Required Debug Effort?

- MultiCore:
  - Business as usual
- ManyCore:
  - More difficult
- DataFlow:
  - Debugging both, application and configuration code

Required Compilation Effort?

- MultiCore/ManyCore:
  - Several minutes
- DataFlow:
  - Several hours for the real hardware.
  - Fortunately, only several minutes for the simulator, several seconds for reload (90% due to DRAM inertia), and several milliseconds to restart the stream and several microseconds to restart the execution.
  - The simulator supports both the large JPMorgan machine as well as the smallest “University Support” machine.
- GoodNews:
  - Tabula@2GHz

Now the Fun Part ☺

Required Space?

- MultiCore:
  - Horse stable

- ManyCore:
  - Chicken house

- DataFlow:
  - Ant hole

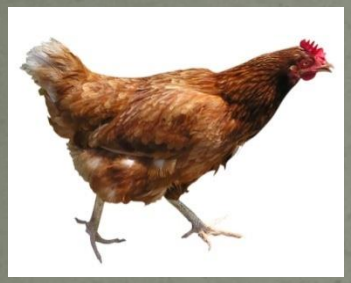
Required Energy?

- MultiCore:
  - Haystack

- ManyCore:
  - Cornbits

- DataFlow:
  - Crumbs

Why Faster?

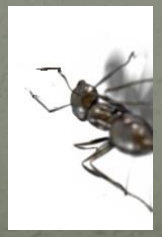


Small Data: Toy Benchmarks (e.g., Linpack)

Why Faster?

Medium Data
(benchmarks
favorising NVidia,
compared to Intel,...)

Why Faster?



Maxeler Dataflow Appliance

- Software based solution
- Dataflow Computing in the Datacentre

MPC C Series

The CPU

Conventional CPU cores and up to 6 DFEs with 288GB of RAM

MPC X Series

The Dataflow Appliance

Dense compute with 8 DFEs, 768GB of RAM and dynamic allocation of DFEs to CPU servers with zero-copy RDMA access

MPC N Series

The Networking Appliance

Intel Xeon CPUs and 4 DFEs with direct links to up to twelve 40Gbit Ethernet connections

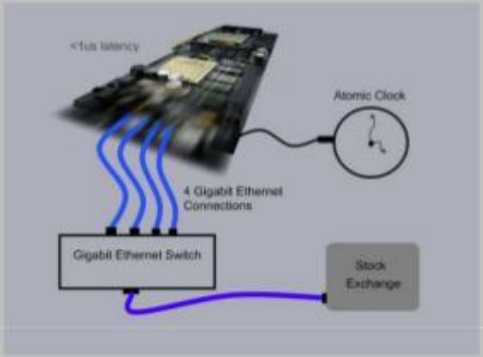
Major Classes of Algorithms, from the Computational Perspective

1. Coarse grained, stateful: Business
   - CPU requires DFE for minutes or hours
   - Interrupts
2. Fine grained, transactional with shared database: DM
   - CPU utilizes DFE for ms to s
   - Many short computations, accessing common database data
3. Fine grained, stateless transactional: Science (Phy, ...)
   - CPU requires DFE for ms to s
   - Many short computations

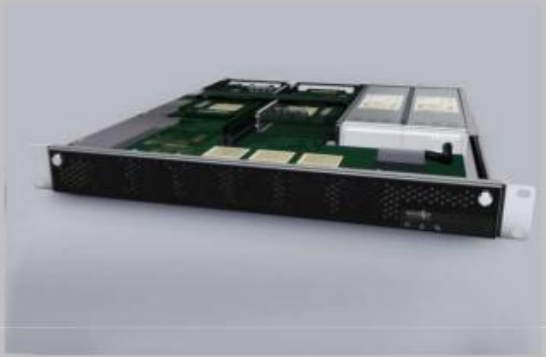
Selected Examples:
Business,
Mathematics,
GeoPhysics, etc.

Maxeler Technologies

MaxCard
e.g. HFT Solution

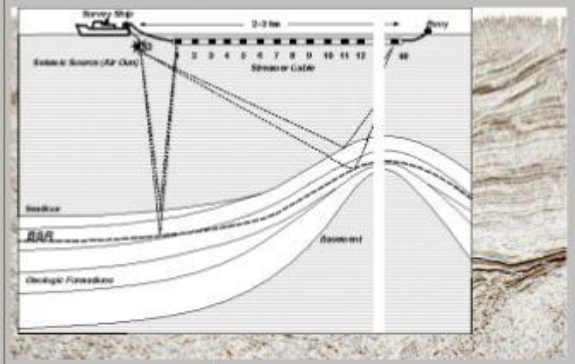


MaxBox
4 MaxCards in a 1U box

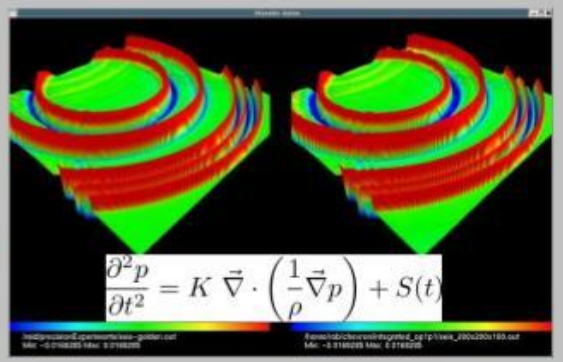


MaxRack
Storage, Network and Compute

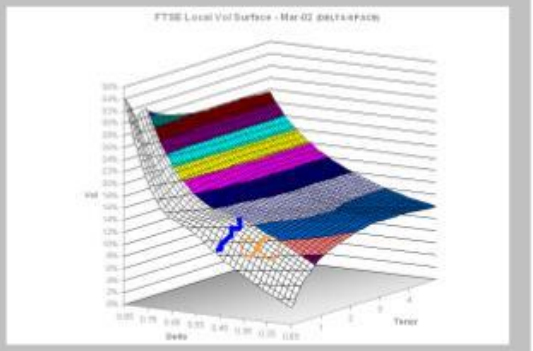
Real-time trace processing



Finite Difference (with Chevron)



Local Vol Approximation



MAXELER Technologies
MAXIMUM PERFORMANCE COMPUTING

An MIS Example: Credit Derivatives

TABLE 3

DISTRIBUTION OF DERIVATIVE CONTRACTS
TOP 25 COMMERCIAL BANKS AND TRUST COMPANIES IN DERIVATIVES
MARCH 31, 2008, $ MILLIONS
NOTE: DATA ARE PRELIMINARY

RANK	BANK NAME	STATE	TOTAL ASSETS	TOTAL DERIVATIVES	PERCENT EXCH TRADED CONTRACTS	PERCENT OTC CONTRACTS	PERCENT INT RATE CONTRACTS	PERCENT FOREIGN EXCH CONTRACTS	PERCENT OTHER CONTRACTS	PERCENT CREDIT DERIVATIVES
					(%)	(%)	(%)	(%)	(%)	(%)
1	JPMORGAN CHASE BANK NA	OH	$1,407,568	89,997,271	5.1	94.9	79.6	8.7	2.7	9.0
2	BANK OF AMERICA NA	NC	1,355,154	37,939,664	4.4	95.6	83.2	7.4	1.2	8.2
3	CITIBANK NATIONAL ASSN	NV	1,292,503	37,691,434	1.8	98.2	75.9	14.3	[illegible]	8.9
4	WACHOVIA BANK NATIONAL ASSN	NC	666,344	4,884,775	9.2	90.8	83.4	4.8	[illegible]	9.3
5	HSBC BANK USA NATIONAL ASSN	DE	188,[illegible]	4,279,737	2.2	97.8	49.9	16.5	2.2	31.3
6	WELLS FARGO BANK NA	SD	[illegible]686	1,440,229	15.1	84.9	94.2	3.4	2.3	0.1
7	BANK OF NEW YORK	NY	128,342	1,058,618	7.5	92.5	82.4	16.4	1.0	0.2
8	STATE STREET BANK&TRUST CO	MA	147,472	904,593	0.3	99.7	3.1	96.[illegible]	0.0	0.0
9	PNC BANK NATIONAL ASSN	PA	128,623	248,705	24.5	75.5	85.4	[illegible]	0.5	2.3
10	SUNTRUST BANK	GA	174,716	241,369	22.6	77.4	90.8	[illegible]	4.7	0.9
11	MELLON BANK NATIONAL ASSN	PA	41,727	192,105	6.0	94.0	20.9	77.7	1.4	0.0
12	NORTHERN TRUST CO	IL	67,962	164,605	0.0	100.0	4.6	95.2	0.0	0.2
13	NATIONAL CITY BANK	OH	152,519	158,612	19.0	81.0	96.3	2.2	0.0	1.4
14			97,979					11.7	0.4	6.4
15			237,269					11.1	0.1	1.7
16			139,766					0.6	0.0	0.3
17			131,916					0.3	0.0	0.6
18			64,564					30.4	0.8	0.5
19			130,820					7.5	0.0	0.5
20			63,003					3.3	3.3	19.8
21			37,064					0.0	0.0	0.0
22			71,098					0.6	0.7	6.2
23			57,413					13.9	13.3	0.0
24	UBS BANK USA	UT	27,989	31,177	0.0	100.0	100.0	0.0	0.0	0.0
25	DEUTSCHE BANK TR CO AMERICAS	NY	38,216	30,693	0.0	100.0	50.9	7.4	26.8	14.9
	TOP 25 COMMERCIAL BANKS & TCs WITH DERIVATIVES		$7,335,274	$179,893,240	$7,970,059	$171,923,180	$141,494,953	$18,466,451	$3,523,248	$16,408,588
	OTHER COMMERCIAL BANKS & TCs WITH DERIVATIVES		2,683,819	450,976	10,934	440,042	369,802	30,984	17,364	32,826
	TOTAL FOR COMMERCIAL BANKS & TCs WITH DERIVATIVES		10,019,092	180,344,216	7,980,994	172,363,222	141,864,755	18,497,435	3,540,612	16,441,414
				(%)	(%)	(%)	(%)	(%)	(%)	(%)
	TOP 25 COMMERCIAL BANKS & TC: % OF TOTAL COMMERCIAL BKS &TCs WITH DERIVATIVES			99.7	4.4	95.3	78.5	10.2	2.0	9.1
	OTHER COMMERCIAL BANKS & TCs: % OF TOTAL COMMERCIAL BKs & TCs WITH DERIVATIVES			0.3	0.0	0.2	0.2	0.0	0.0	0.0
	TOTAL FOR COMMERCIAL BANKs & TCs: % OF TOTAL COMMERCIAL BANKs & TCs WITH DERIVATIVES			100.0	4.4	95.6	78.7	10.3	2.0	9.1

Note: Currently, the Call Report does not differentiate credit derivatives by over the counter or exchange traded. Credit derivatives have been included in the "over the counter" category as well as in the sum of total derivatives here.
Note: "Foreign Exchange" does not include spot fx.
Note: "Other" is defined as the sum of commodity and equity contracts.
Note: Numbers may not add due to rounding.
Data source: Call Reports, schedule RC-L

J.P. Morgan 89.997 Trillion Derivatives

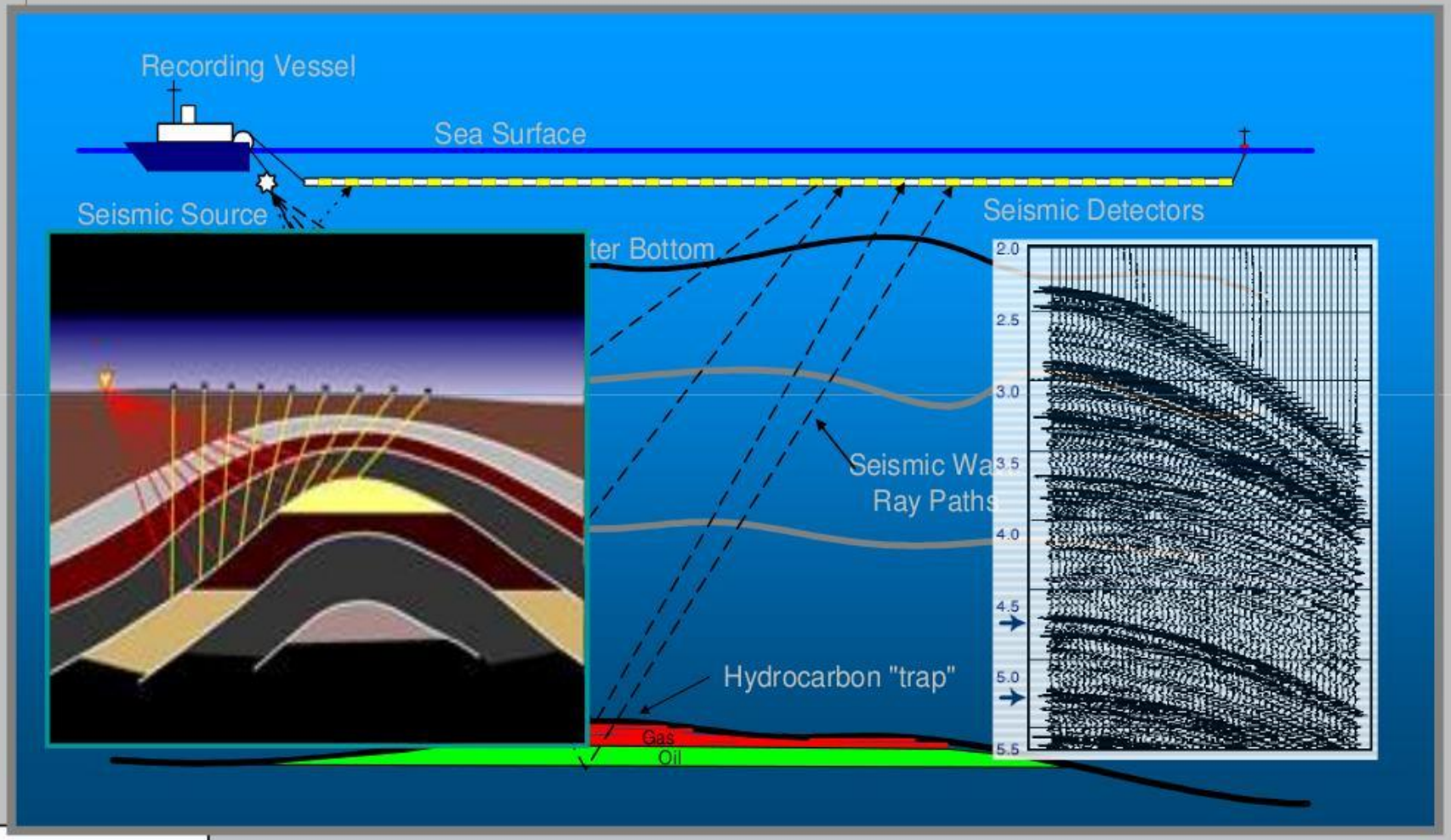
9 % of book is credit derivatives

Fujii Laboratory

Dynamics and Control of Space Infrastructures

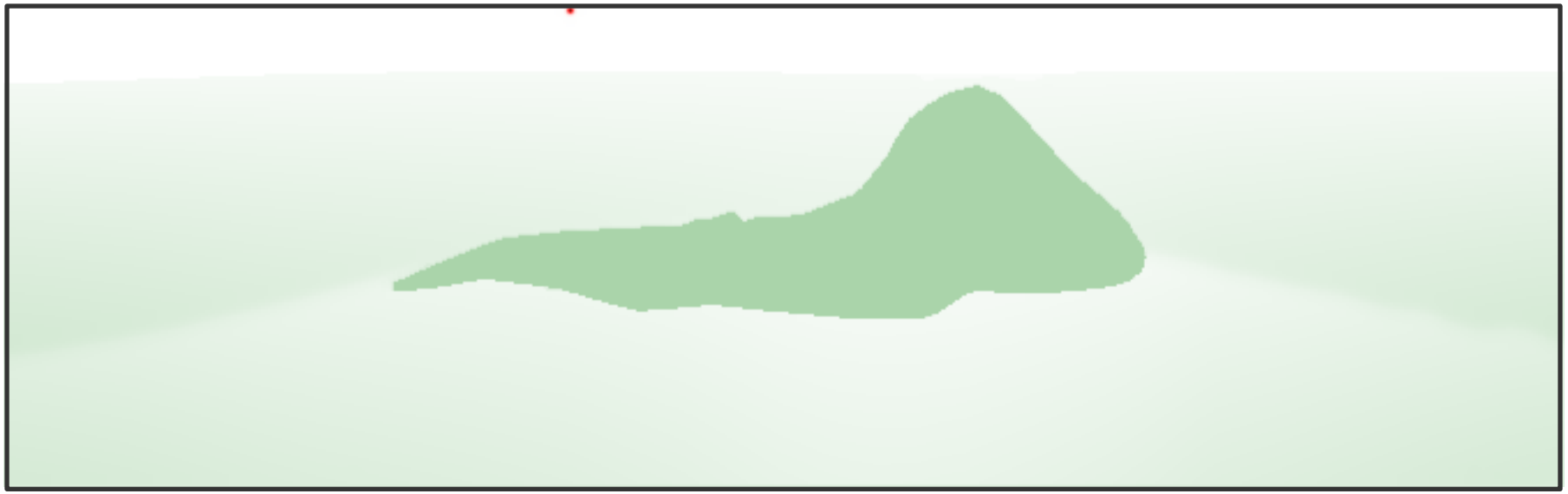
HW

Seismic Data Acquisition



Courtesy of Schlumberger

Seismic Imaging



- Running on MaxNode servers
  - 8 parallel compute pipelines per chip
  - 10x less power: 150MHz vs 1.5GHz
  - 30x faster than microprocessors

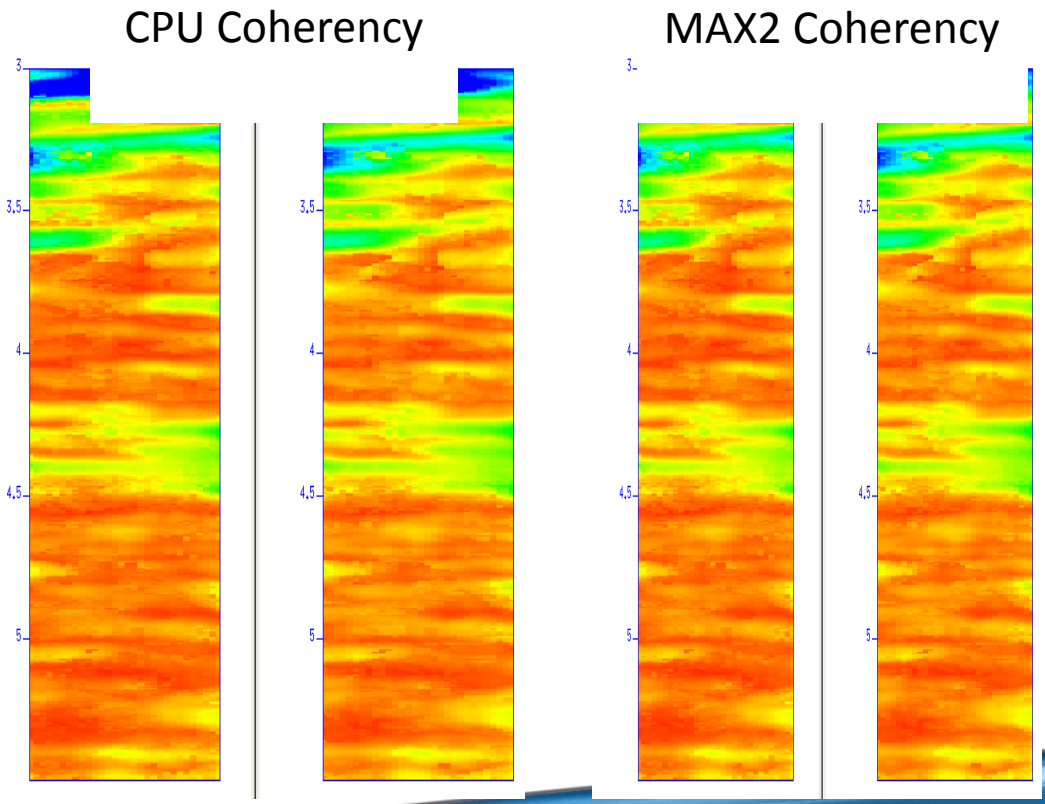
An Implementation of the Acoustic Wave Equation on FPGAs

T. Nemeth[†], J. Stefani[†], W. Liu[†], R. Dimond[‡], O. Pell[‡], R.Ergas[§]

[†]Chevron, [‡]Maxeler, [§]Formerly Chevron, SEG 2008

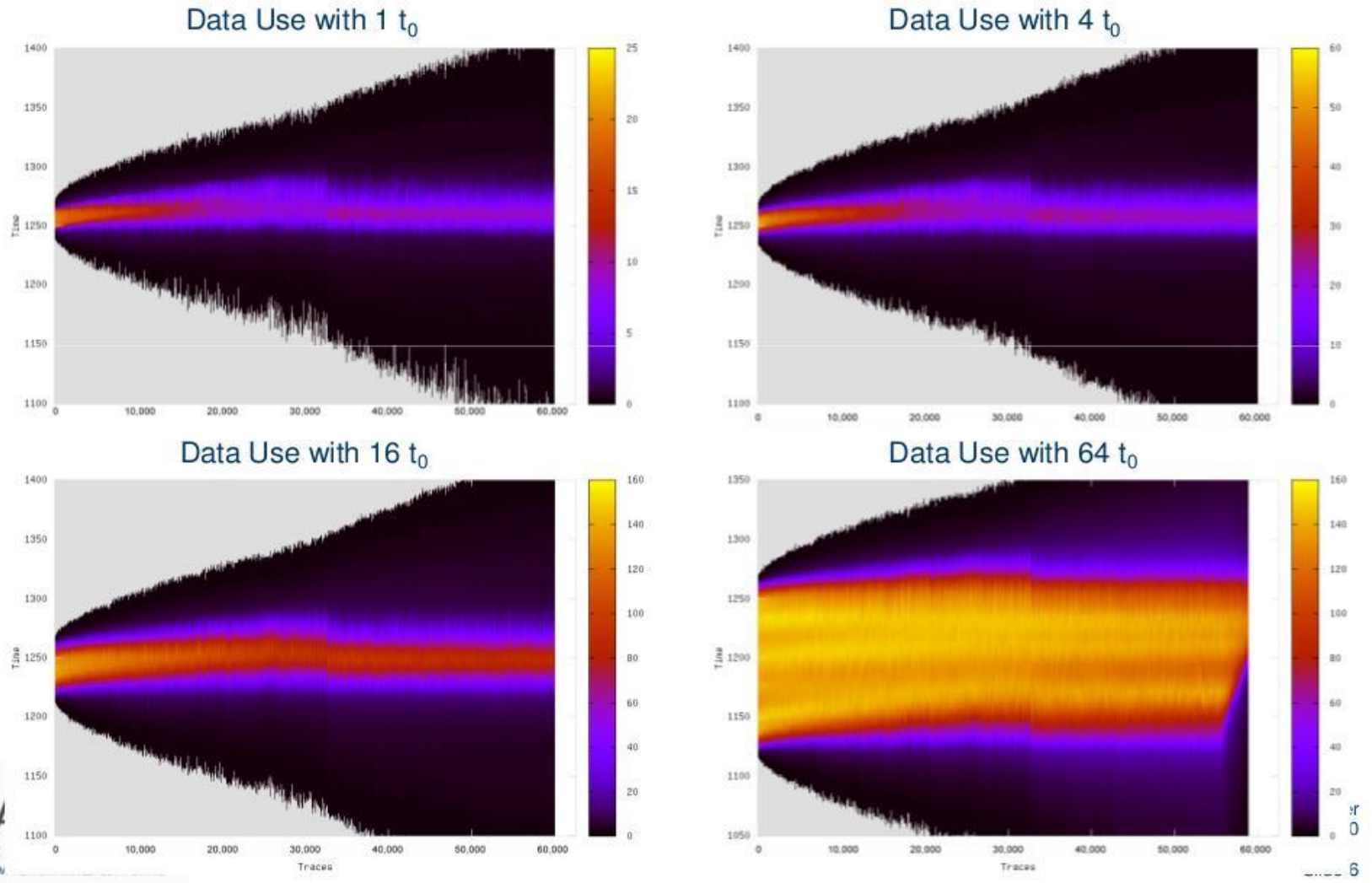
The CRS Results

- Performance of one MAX2 card vs. 1 CPU core
  - Land case (8 params), speedup of 230x
  - Marine case (6 params), speedup of 190x



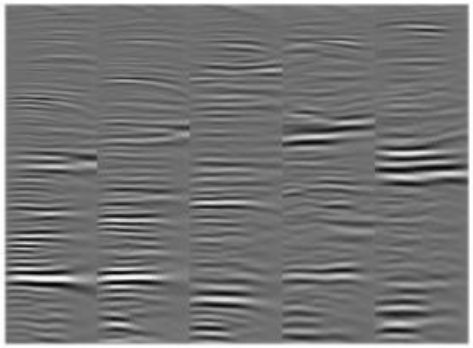
ENI-AGIP Seismic Trace App: Conjugate Gradient Optimization

100 MAX2 cards delivering performance of 21,800 CPU cores[EAGE2010]



48x Speedup of Angle Gathers

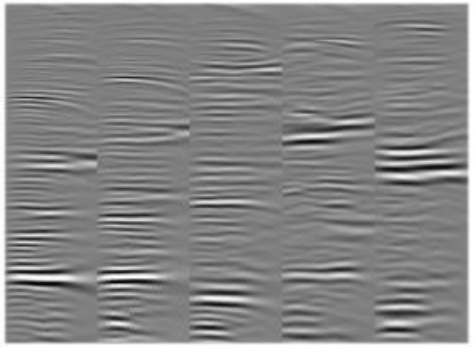
with Stanford Center for Earth and Environmental Sciences *)



Angle gathers from CPU computed subsurface offsets

- Can be dominant cost in shot profile migration
- Cross-correlating two fields by various shifts:

$$I(h,x,z) = \sum_{s} \sum_{w} S(x-h,z,w,s) \cdot G^{*}(x+h,z,w,s)$$



Angle gathers from FPGA computed subsurface offsets

SPEEDUP RESULTS FROM CUSTOM TRACE MEMORY SYSTEM:

- **Trace = Unit of Transfer**
- **Buffers Prefetch Right Traces in Advance**

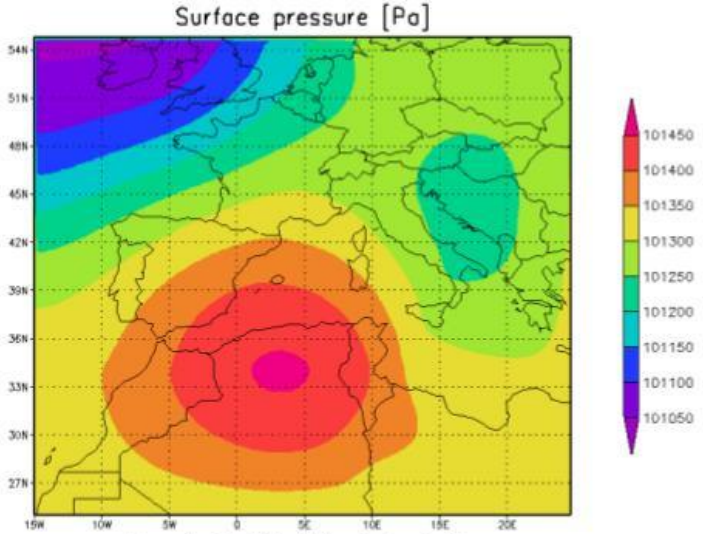
MAXELER Technologies
MAXIMUM PERFORMANCE COMPUTING

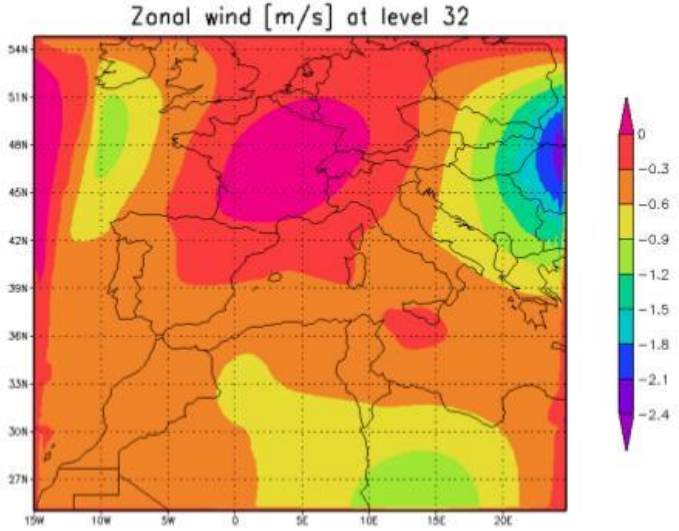
*) Case Study: Subsurface Offset Gathers, presented at SEG 2008

More Results

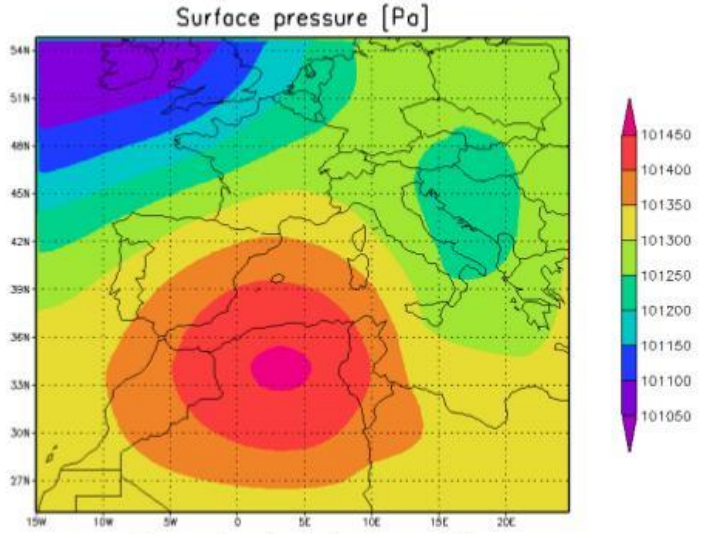
Problem size: (Longitude) 13,600 Km x (Latitude) 3330 Km
Simulation of baroclinic instability after 2 hrs (500 time steps)

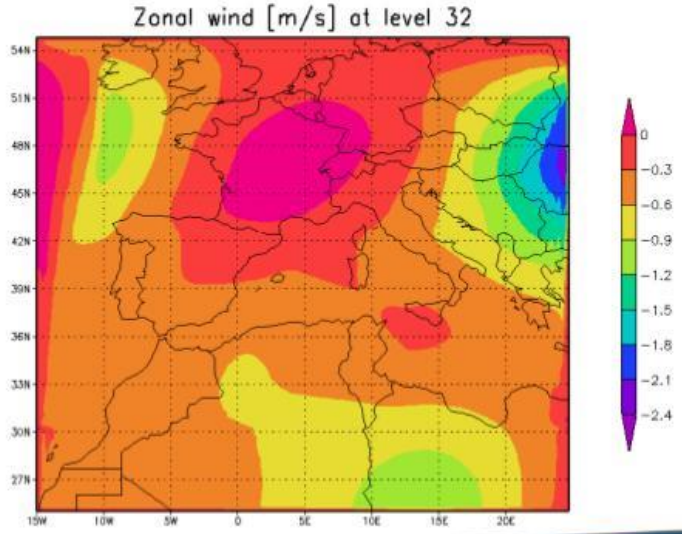
CPU





Dataflow





MAXELER Technologies

Trace Stacking: Speed-up 217

- DM for Monitoring and Control in Seismic processing
- Velocity independent / data driven method to obtain a stack of traces, based on 8 parameters
- Search for every sample of each output trace

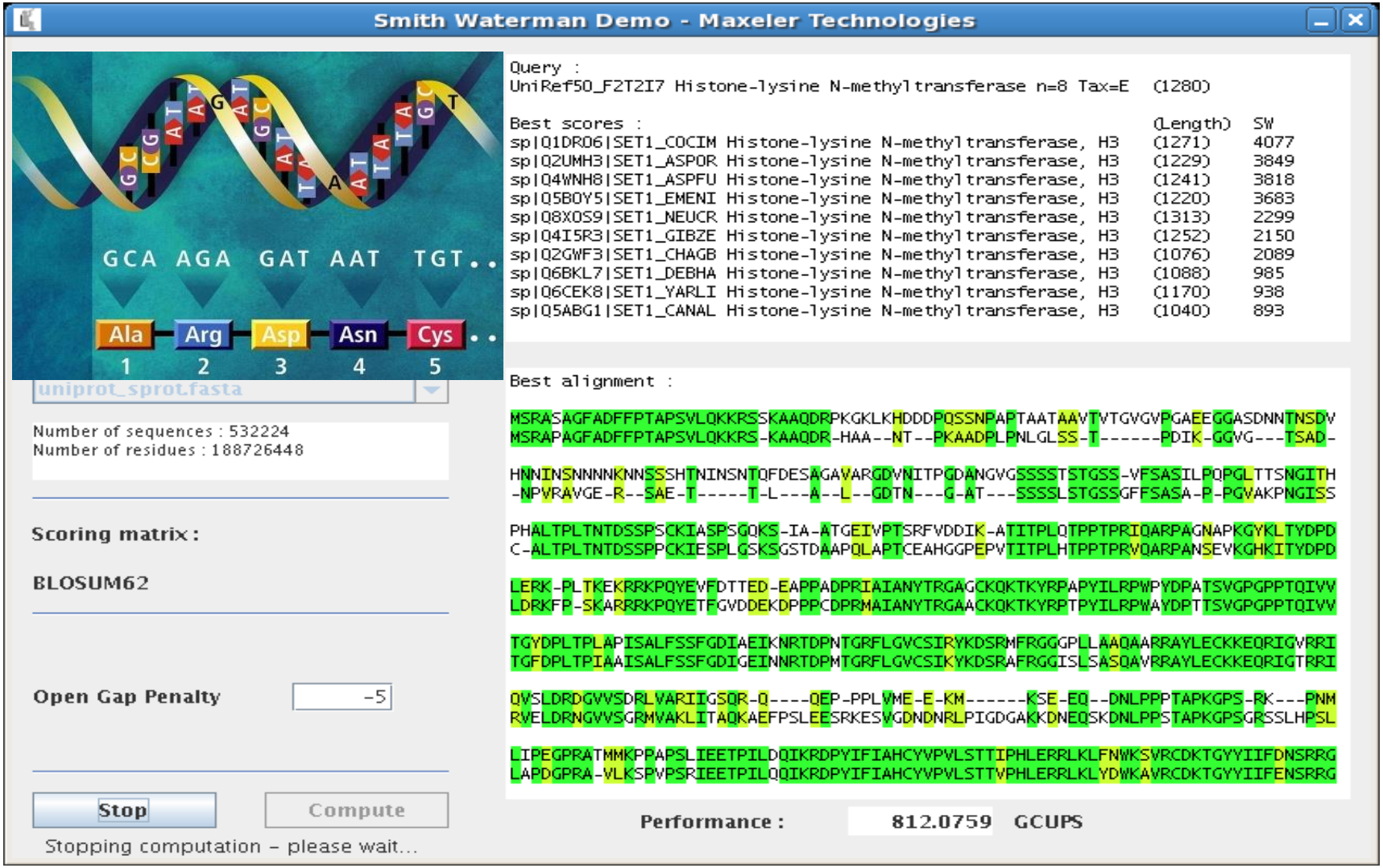
$$t_{hyp}^2 = \left(t_0 + \frac{2}{v_0} \mathbf{w}^T \mathbf{m} \right)^2 + \frac{2t_0}{v_0} \left(\mathbf{m}^T \mathbf{H}_{zy} \mathbf{K}_N \mathbf{H}_{zy}^T \mathbf{m} + \mathbf{h}^T \mathbf{H}_{zy} \mathbf{K}_{NIP} \mathbf{H}_{zy}^T \mathbf{h} \right)$$

2 parameters (emergence angle & azimuth)

3 Normal Wave front parameters ($K_{N,11}$; $K_{N,12}$; K_{N22})

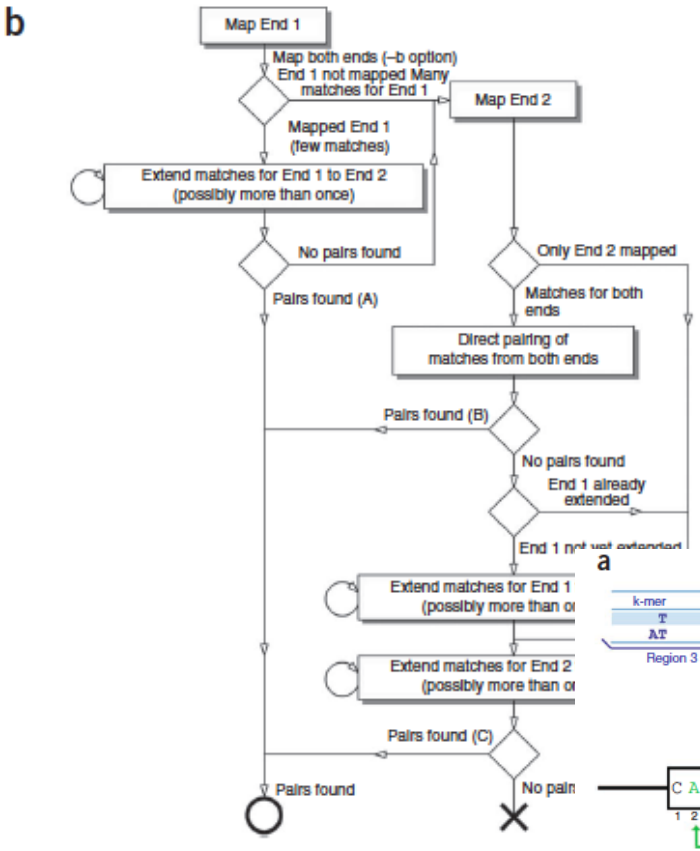
3 NIP Wave front parameters ($K_{Nip,11}$; $K_{Nip,12}$; K_{Nip22})

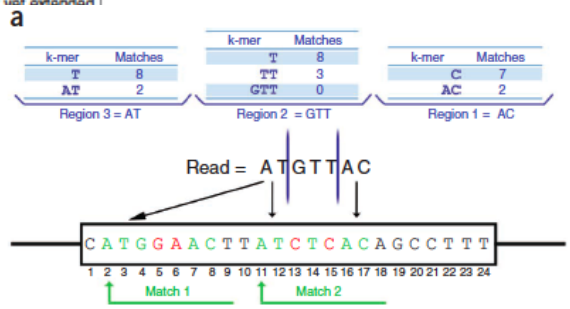
Maxeler Running Smith Waterman

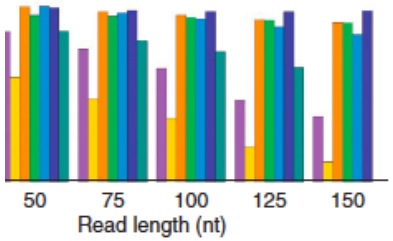


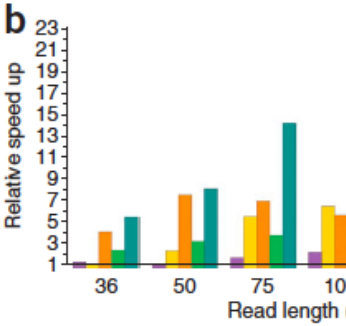
The GEM mapper: fast, accurate and versatile alignment by filtration

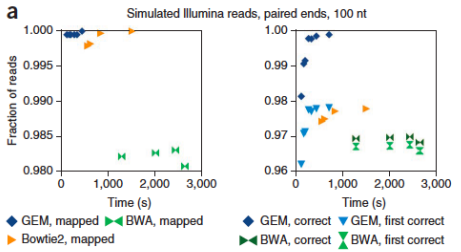
Santiago Marco-Sola[1], Michael Sammeth[1], Roderic Guigó[2] & Paolo Ribeca[1,3]

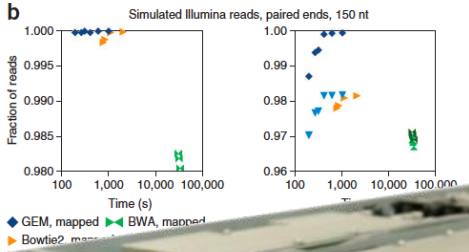


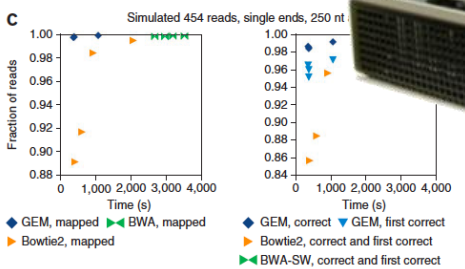


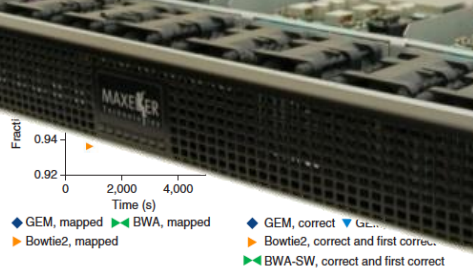


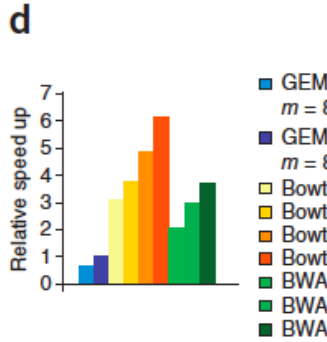






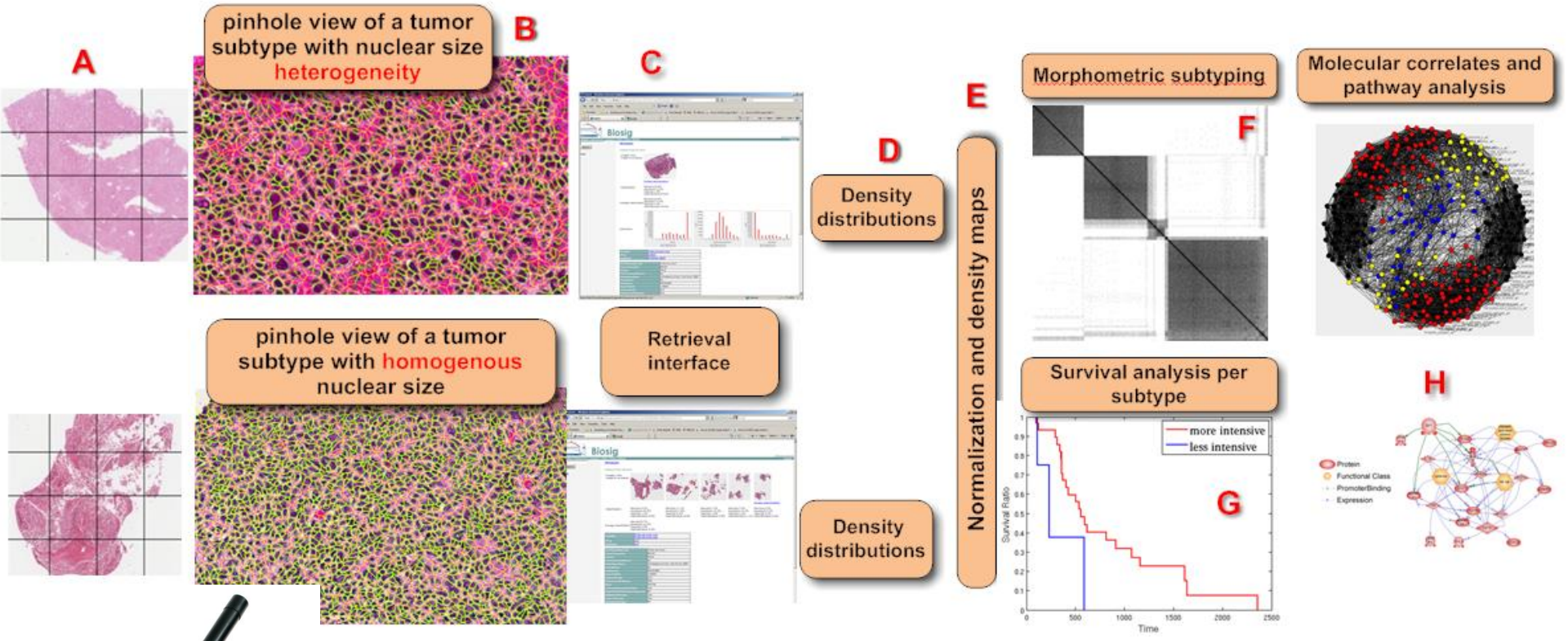






Molecular Correlates of Tumor Signatures from a Large Cohort

From whole slide sections, of a cohort, to pathway analysis (Prof Bahram Parvin, Berkeley)



High Content Analysis (HCA) on MPC-X

MAXELER Technologies

Acceleration is Hard

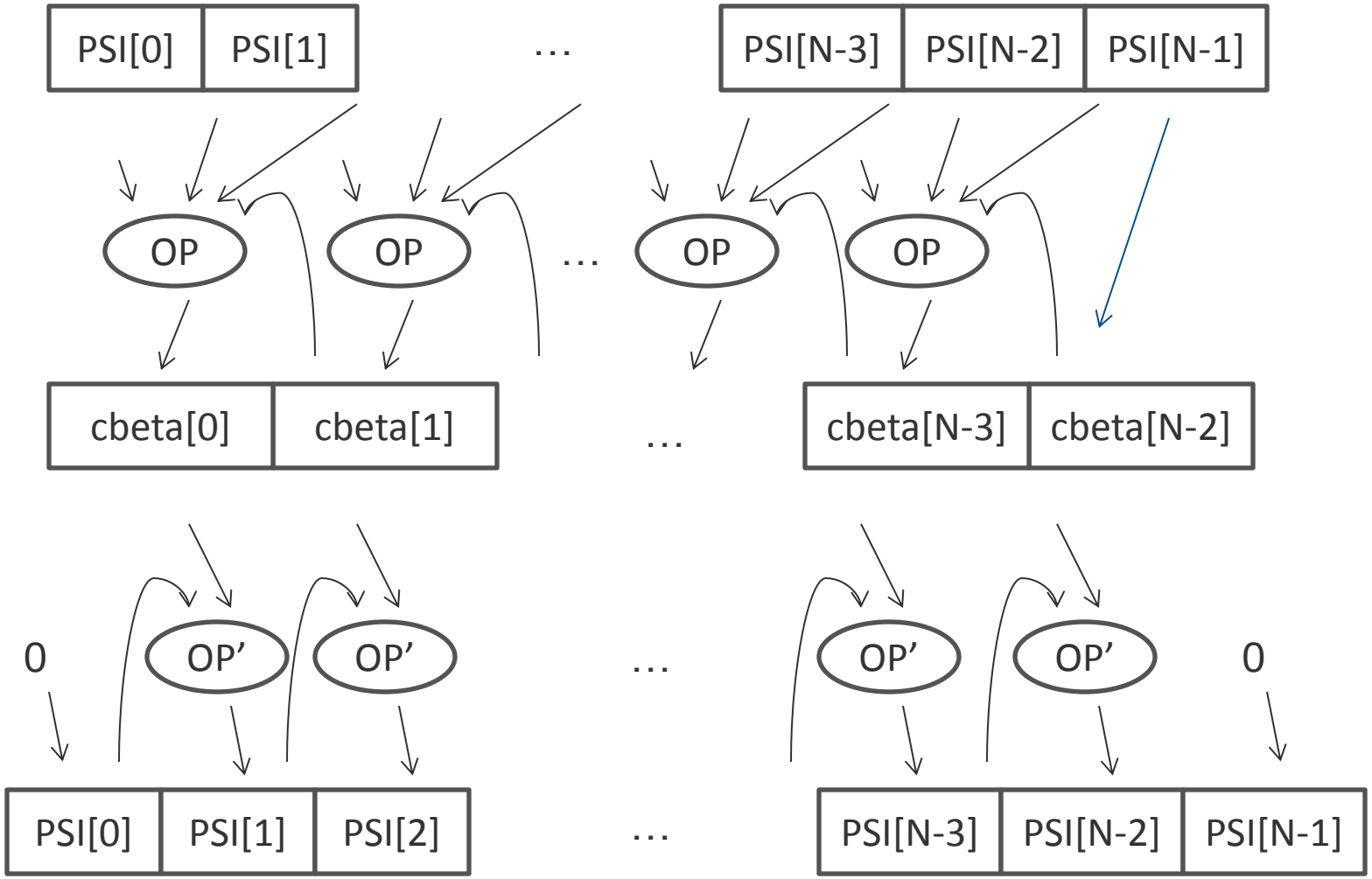
Conclusion: Nota Bene

This is about algorithmic changes,
to maximize
the algorithm to architecture match:

algorithmic modifications,
pipeline utilization,
data choreography,
and
decision making precision.

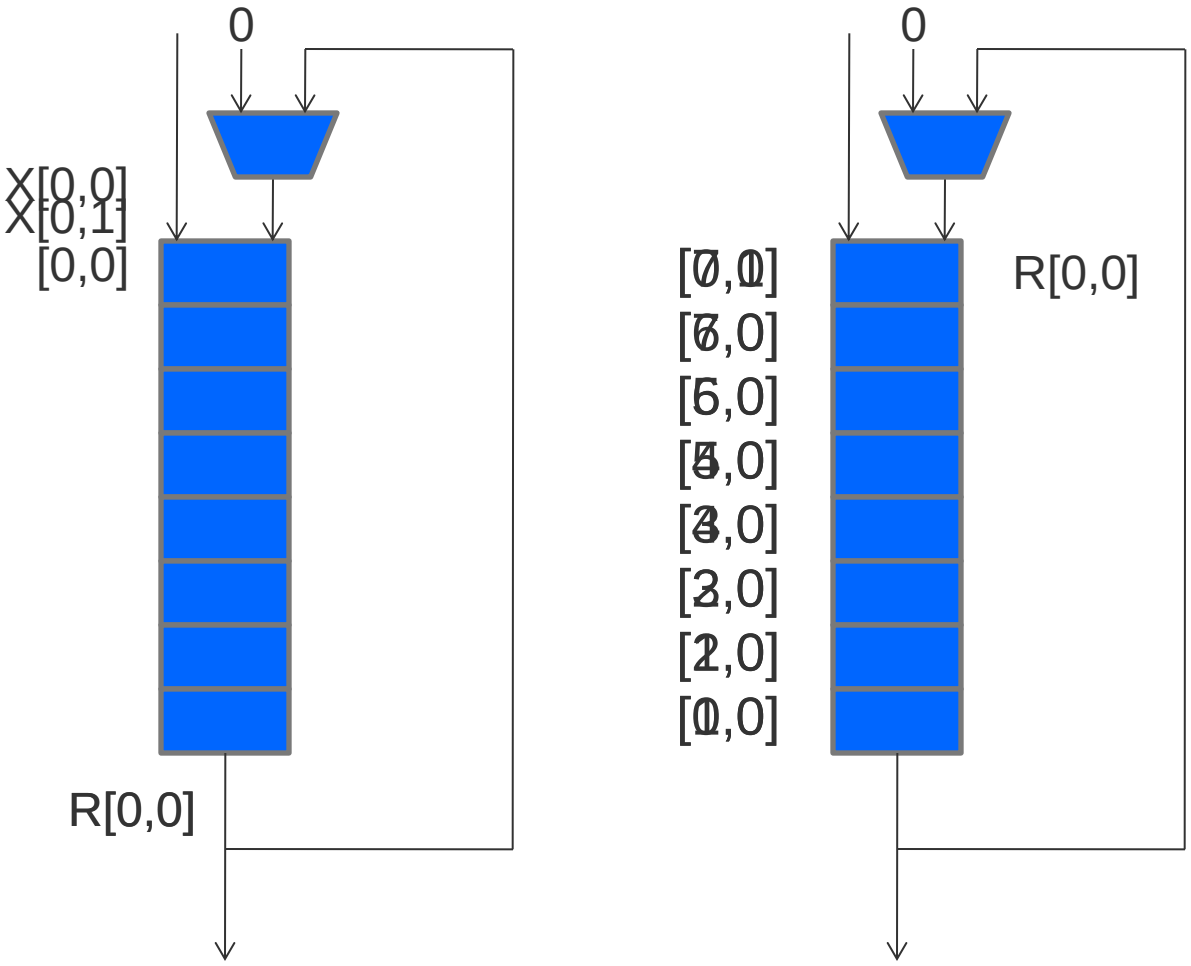
The winning paradigm of Big Data ExaScale?

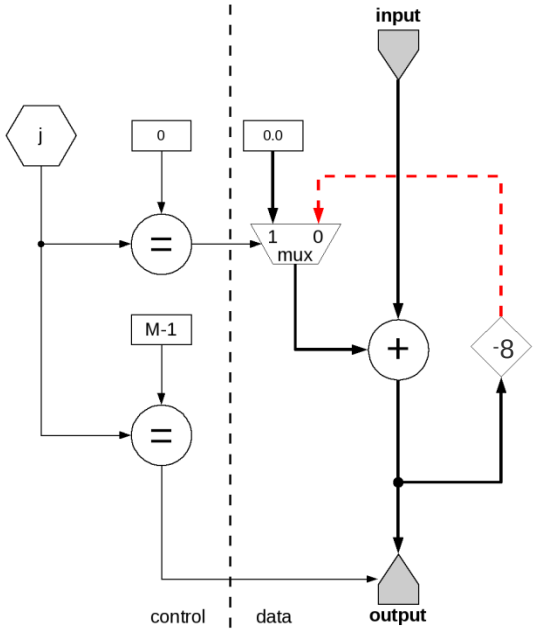
Algorithmic Changes: Data Dependencies



Example generated by Sasa Stojanovic (Gross-Pitaevskii)

Pipeline Changes: Higher Efficiency

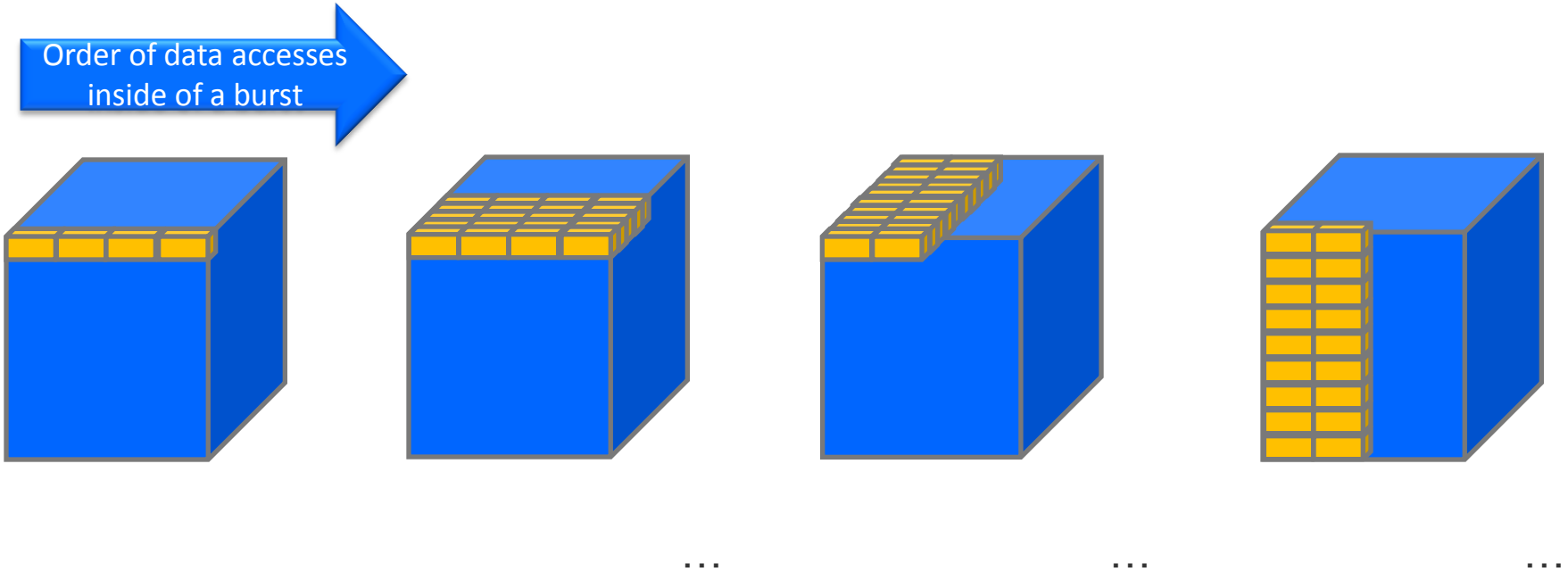




Example generated by Sasa Stojanovic (Gross-Pitaevskii)

Data Recoreography: Pipeline Utilization

Example generated by Sasa Stojanovic (Gross-Pitaevskii)



Fixed Point: Savings Reinvestable

- Consider fixed point
  compared to single precision floating point
- If the range is tightly confined,
  one could use *24-bit* fixed point
- If data has a wider range, may need *32-bit* fixed point

	hwFloat(8,24)	hwFix(24,...)	hwFix(32,...)
Add	500 LUTs	24 LUTs	32 LUTs
Multiply	2 DSPs	2 DSPs	4 DSPs

- Arithmetic is not 100% of the chip.
  In practice, often ~5x performance boost from fixed point.

DataFlow for ExaScale DM

- Revisiting the Top 500 SuperComputers benchmarks
  - Our paper in Communications of the ACM
- Revisiting all major Big Data DM algorithms
  - Massive static parallelism at low clock frequencies
- Concurrency and communication
  - Concurrency between millions of tiny cores difficult, "jitter" between cores will harm performance at synchronization points
- Reliability and fault tolerance
  - 10-100x fewer nodes, failures much less often
- Memory bandwidth and FLOP/byte ratio
  - Optimize **data choreography, data movement**, and the algorithmic computation
- New architecture of n-Programming paradigms

FP7: RoMoL@BCN

Mateo Valero

Mateo Valero Cortés is a Spanish computer architect. Valero received the Eckert-Mauchly award in 2007, for "extraordinary leadership in building a world class computer architecture research center, for ... Wikipedia

Education: Polytechnic University of Catalonia

Awards: Eckert–Mauchly Award

The SAB goal: Out of box thinking!

FP7: BalCon@SRB

The vision of Alkis Konstantellos

The SAB goal: Seed for new proposals!

DAFNE: Leader MISANU

DAFNE = South (MaxCode) + North (BigData)

MISANU, IMP, KG, NS,	UK
BSC, UPV,	Sweden
U of Siena, U of Roma,	Norway
IJS, FRI,	Denmark
IRB,	Germany
QPLAN,	France
Bogazici, U of Istanbul,	Austria
U of Bucharest, U of Arad,	Swiss
U of Tuzla,	Poland
Technion, Maxeler Israel, IPSI	Hungary

The DAFNE Map

The TriPeak @ DATAMAN

Siena
+ BSC
+ Imperial College
+ Maxeler
+ Belgrade

The TriPeak: Essence

MontBlanc = A ManyCore (NVidia) + a MultiCore (ARM)
Maxeler = A FineGrain DataFlow (FPGA)

How about a happy marriage?
MontBlanc (ompSS) and Maxeler (an accelerator)

In each happy marriage,
it is known who does what :)

The Big Data DM algorithms:
What part goes to MontBlanc and what to Maxeler?

TriPeak: Core of the Symbiotic Success

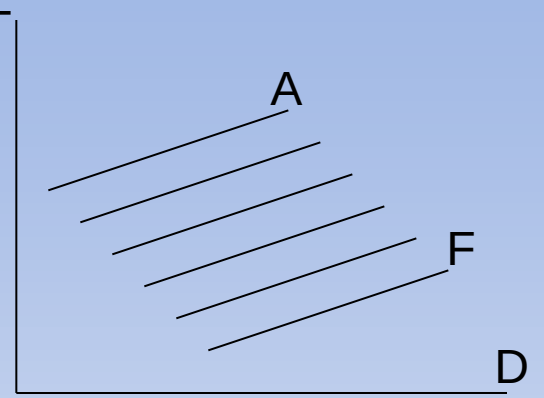
An intelligent DM algorithmic scheduler,
partially implemented for compile time,
and partially for run time.

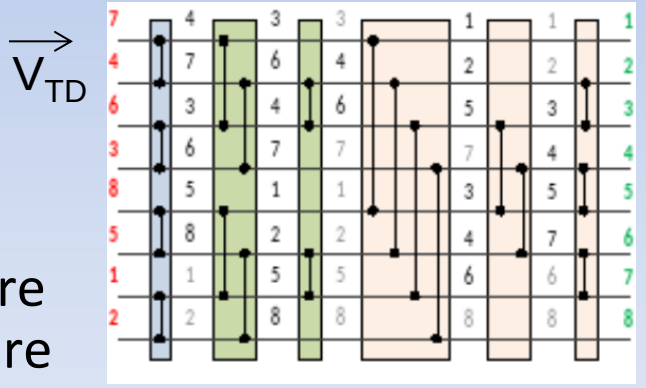
At compile time:
Checking what part of code fits where
(MontBlanc or Maxeler): LoC 1M vs 2K vs 20K

At run time:
Rechecking the compile time decision,
based on the current data values.

Maxeler: Research (Google: good method)

Structure of a Typical Research Paper: Scenario #1
[Comparison of Platforms for One Algorithm]
Curve A: MultiCore of approximately the same price
Curve B: ManyCore of approximately the same price
Curve C: Maxeler after a direct algorithm migration
Curve D: Maxeler after algorithmic improvements
Curve E: Maxeler after data choreography
Curve F: Maxeler after precision modifications





Structure of a Typical Research Paper: Scenario #2
[Ranking of Algorithms for One Application]
CurveSet A: Comparison of Algorithms on a MultiCore
CurveSet B: Comparison of Algorithms on a ManyCore
CurveSet C: Comparison on Maxeler, after a direct algorithm migration
CurveSet D: Comparison on Maxeler, after algorithmic improvements
CurveSet E: Comparison on Maxeler, after data re-choreography
CurveSet F: Comparison on Maxeler, after precision modifications

Maxeler Research in Serbia: Special Issue of IPSI Transactions Journal

KG: Blood Flow, Tijana Djukic and Prof. Filipovic

NS: Combinatorial Math, Prof. Senk and Ivan Stanojevic

MISANU: The SAT Math, Zivojin Sustran and Prof. Ognjanovic

ETF: Meteorology, Radomir Radojicic and Marko Stankovic

ETF: Physics (Gross Pitaevskii 3D real), Sasa Stojanovic

ETF: Physics (Gross Pitaevskii 3D imaginary), Lena Parezanovic

The Early Bird Teaching Efforts

Stanford (Michael Flynn)
arith.stanford.edu/~flynn

Imperial College of London (Oskar Mencer)
cc.doc.ic.ac.uk/opensp114

Universty of Belgrade (Veljko Milutinovic)
home.etf.rs/~vm/os/vlsi/predavanja/maxeler.html

Maxeler: Teaching (Google: prof vm)

TEACHING, VLSI, PowerPoints: MAXELER, http://home.etf.rs/~vm/os/vlsi/predavanja/maxeler.html

Maxeler Veljko Anegdotic PPT PDF
Paper by Jakob Salom
A Paper from Tsinghua on Shallow Water Weather Forecast
Tutorial Slides by Sasha and Veljko: Practice (Current Update)
Maxeler Oskar Introduction, 2012
Maxeler Oskar Prediction, 2018
Tutorial Slides by Oskar: Theory (6 parts)
Slides by Jacob, New York
Slides by Jacob, Alabama
Slides by Sasha: Practice (Current Update)
Maxeler in Meteorology
Maxeler in Mathematics
Maxeler Related Work of PHD Students
Maxeler Essence by Korolija
Maxeler Belgrade Veljko Summary, August 2012
Discusion of Two Possible Research Avenues
Maxeler Forbes Article
Maxeler JPMorgan Flyer
Homework Solutions (6 parts)
Catalog
Brochure
OpenSPL
Oskar@Linz Keynote Speach
Maxeler@USA Course Logistics
Maxeler Tutorial: The Full Blown Text
Paper, unconditionally accepted for Advances in Computers by Elsevier (March 2015)
Paper, unconditionally accepted for Communications of the ACM (May 2013)
webide.maxeler.com maxeler.mi.sanu.ac.rs maxIDE@MISANU
The IET Best Paper 2014 Premium Award for Computing and Digital Techniques by Jovanovic and Milutinovic
The IEEE/ACM HICSS Best Paper Award for The Architecture Track by Fortes and Milutinovic
AppGallery.maxeler.com
The Newest March 2015 Book by Elsevier, on DataFlow Concepts and Applications: Dataflow Processing (Advances in Computers, Volume 96)
The Newest May 2015 Book by Springer, on DataFlow Concepts and Applications: Guide to DataFlow SuperComputing (Computer Communications and Networks)
www.maxeler.com/quotation
Class Project: Numerical Receipts in MaxJ
Class Project: Functions of the Language R in MaxJ
Google Giving Up MapReduce in Favor of DataFlow Intel Trying to Acquire Altera

THE COURSE ALSO INCLUDES DARPA METHODOLOGY FOR MICROPROCESSOR DESIGN

Maxeler PreConference Tutorials (2013)

Google:

IEEE HiPeak, Berlin, Germany, January 2013

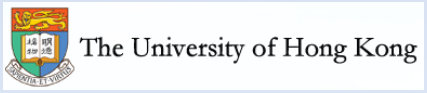
ACM iSAC, Coimbra, Portugal, March 2013

IEEE MECO, Budva, Montenegro, June 2013

ACM ISCA, Tel Aviv, Israel, June 2013

Maxeler InHouse Tutorials (2015)

Maxeler University Program Members



How to Become a Family Member?

Galava: Highly capable PCI-e dataflow compute card, available for Universities to enable affordable dataflow acceleration.

Here you can request a Quotation for our **GALAVA DFE** in the context of the Maxeler University Program (MAX-UP). You can fill in the quotation **Google Form**.

Features

- Programmable logic fabric with 490,000 elements
- 500 programmable multipliers
- 5.6 MB of on-chip Fast Memory (FMEM)
- 12 GB DDR3 DRAM Large Memory (LMEM)
- PCI-e link with 2GB/s (peak) bandwidth

Requirements

- Linux workstation running CentOS (v.6.5)
- Two free PCIe slots located side-by-side
- One free 6-pin flat connector from the power supply
- space for **full height x 3/4 length** card (111,125 x 247 mm)

Options to consider:

a. Applying for MAX-UP free of charge

b. Purchasing a university-level machine (about $5K)

c. Purchasing a JPM-level machine
(the value approaching $100M),
or at least a Schlumberger-level machine
(the value moving above $10M)

Our Work Impacting Our Interest in Maxeler

Milutinovic, V., Knezevic, P., Radunovic, B., Casselman, S., Schewel, J., Obelix Searches Internet Using Customer Data, IEEE COMPUTER, July 2000 (impact factor 2.205/2010).

Milutinovic, V., Cvetkovic, D., Mirkovic, J., Genetic Search Based on Multiple Mutation Approaches, IEEE COMPUTER, November 2000 (impact factor 2.205/2010).

Milutinovic, V., Ngom, A., Stojmenovic, I., STRIP --- A Strip Based Neural Network Growth Algorithm for Learning Multiple-Valued Functions, IEEE TRANSACTIONS ON NEURAL NETWORKS, March 2001, Vol.12, No.2, pp. 212-227. IEEE Awarded

Jovanov, E., Milutinovic, V., Hurson, A., Acceleration of Nonnumeric Operations Using Hardware Support for the Ordered Table Hashing Algorithms, IEEE TRANSACTIONS ON COMPUTERS, September 2002, Vol.51, No.9, pp. 1026-1040 (impact factor 1.822/2010). The IEEE/ACM 1997 Architecture Track Best Paper Award

Maxeler Impacting Our Work

Tafa, Z., Rakocevic, G., Mihailovic, Dj., Milutinovic, V., Effects of Interdisciplinary Education On Technology-driven Application Design IEEE Transactions on Education, August 2011, pp.462-470. (impact factor 1.328/2010).

Tomazic, S., Pavlovic, V., Milovanovic, J., Sodnik, J., Kos, A., Stancin, S., Milutinovic, V., Fast File Existence Checking in Archiving Systems ACM Transactions on Storage (TOS) TOS Homepage archive, Volume 7 Issue 1, June 2011, ACM New York, NY, USA.

Jovanovic, Z., Milutinovic, V., FPGA Accelerator for Floating-Point Matrix Multiplication, IEE Computers & Digital Techniques, 2012, 6, (4), pp. 249-256. The IET 2014 Premium Award for Computing & Digital Techniques

Flynn, M., Mencer, O., Milutinovic, V., Rakocevic, G., Stenstrom, P., Trobec, R., and Valero, M., Moving from Petaflops (on Simple Benchmarks) to Petadata per Unit of Time and Power (On Sophisticated Benchmarks) Communications of the ACM, May 2013 (impact factor 1.919/2010) . ACM Awarded

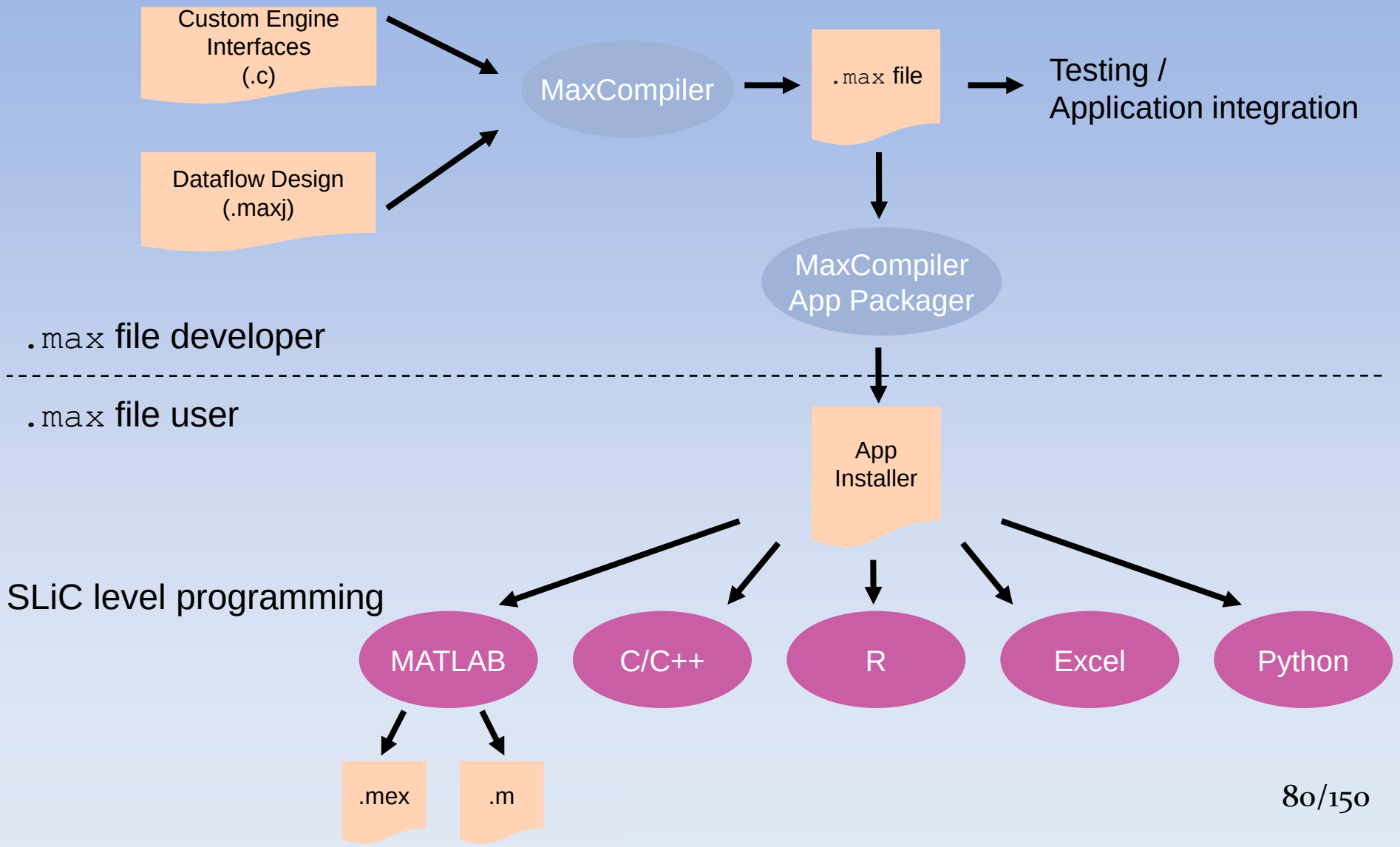
Current Main Efforts of Maxeler

1. To encourage a lot of software to be written/ported.
   This is a key business opportunity that needs to be developed.
2. Maxeler is building up a website and a community
   to share software for DFEs.
   This would allow the software to also be sold
   directly from the Maxeler website.
3. If a PhD student ports an important software
   to a Maxeler machine,
   she/he could become the first software vendor in the world
   for dataflow computers,
   and Maxeler would be happy to help sell licenses.

Current Side Efforts of Maxeler

1. Developing new tools for easier making of kernels.
2. Bringing new languages to Maxeler:
   C, C++, MathLab, Matematika
3. Porting popular application packages to Maxeler:
   OpenSees, etc...
4. Trying the Tabula FPGA!
5. Getting more than 1TeraByte/sec thru I/O
6. Minimizing the hardware, so it can go into Galaxy 5,6…

NewTools: MaxSkins



Getting Started a Practical Work from the Linux Shell

1. Open a shell terminal (e.g., $ /usr/bin/xfce4-terminal).
2. Connect to the Maxeler machine
   (e.g., $ ssh root@147.91.12.216).
3. If more shell screens needed, start screen (e.g., $ screen).
4. Switch to the directory that contains
   the 2n+3 programs you wrote
   (e.g., $ cd Desktop/workspace/src/ind/z88/).
5. Prepare your C code for measuring the execution time
   (e.g., clock_gettime(CLOCK_REALTIME, &t2);).
6. See what you can do (e.g., $ make).
7. Select one of those that you can do
   (e.g., $ make build-sim, $ make run-sim,
   $ make build-hw, $ make run-hw).
8. Measure the power consumption at the wall plug.

New CME Electronic Trading Gateway:

Currently the World's Largest Futures and Options Exchange

New iLink Architecture

New Architecture and Network Enhancements

Program Goals

•Extend CME Group's technology leadership with the first FPGA-based exchange order entry gateway to improve predictability and consistency, especially for order entry bursts

- FPGA technology provided by Maxeler Technologies

•Extend First In, First Out (FIFO) message handling from the trading engine to the order entry gateway eliminating the need to focus on performance variability across sessions

•Harden risk management with advanced market controls including new inline, pre-trade credit controls aggregated down to the account and product level

Scope

•New iLink order routing network and architecture

•New order entry gateways aligned with Market Segments

- Market Segment Gateways (MSGW) with FIFO message handling
- Customers responsible for routing orders to correct MSGW

•Current, market segment-agnostic iLink gateways are now referred to as Convenience Gateways (CGW)

- Convenience Gateways route orders to the correct MSGW

•New session identifiers and access model

•Session policy updates

Live as of March 2014

CME Group

Designed for educational use only using Maxeler Technologies' curve construction methodology. This tool uses delayed data and displayed results are indicative representations only.

Please hover your mouse pointer over column titles and links for further information.

CME Ticker	Bloomberg Ticker	DSF Pricing					Timestamp
		Price	Coupon	PV01	NPV	Implied Rate	
T1UM4 2Y	CTPM4	100'057	0.750%	$19.97	$179.69	0.6600%	4:00:03 PM CT 4/4/2014
F1UM4 5Y	CFPM4	100'115	2.000%	$48.49	$359.38	1.9259%	4:00:03 PM CT 4/4/2014
N1UM4 10Y	CNPM4	100'225	3.000%	$90.16	$703.12	2.9220%	4:00:03 PM CT 4/4/2014
B1UM4 30Y	CBPM4	102'270	3.750%	$195.07	$2,843.75	3.6042%	4:00:03 PM CT 4/4/2014
T1UU4 2Y	CTPU4	100'085	1.000%	$19.93	$265.62	0.8668%	4:00:03 PM CT 4/4/2014
F1UU4 5Y	CFPU4	100'110	2.250%	$48.27	$343.75	2.1788%	4:00:03 PM CT 4/4/2014
N1UU4 10Y	CNPU4	101'125	3.250%	$89.55	$1,390.62	3.0948%	4:00:03 PM CT 4/4/2014
B1UU4 30Y	CBPU4	106'020	4.000%	$193.47	$6,062.50	3.6868%	4:00:03 PM CT 4/4/2014

Quotes and analytics are updated every 15 minutes.

Analytics powered by Maxeler Technologies©

Instrument	CPU 1U-Node	Max 1U-Node	Comparison
European Swaptions	848,000	35,544,000	42x
American Options	38,400,000	720,000,000	19x
European Options	32,000,000	7,080,000,000	221x
Bermudan Swaptions	296	6,666	23x
Vanilla Swaps	176,000	32,800,000	186x
CDS	432,000	13,904,000	32x
CDS Bootstrap	14,000	872,000	62x

Dataflow Computing at UK Government

Industry

Government

Engagement type

Hardware and Software Platform Sale

Main Contacts

Ministry of Science

Date

Dec 2013

Critical Client Issues

- Competitive advantage in international race
- Transitioning to Big Data Analytics while conventional solutions do not manage to keep up
- High Energy Physics keeps pushing computational demands

Approach

The approach:

1. Selecting Multiscale Dataflow Computing architecture
2. Training of staff to work with Maxeler technology
3. Investment into Dataflow Software development
4. Support of initial users in the UK and across Europe

Client benefits

- 20-50x increased compute capability per cubic-foot of data center space
  => single Maxeler rack brings compute capability of over 20 conventional racks
- Enabling the evaluation of portable Petascale computing systems
- Green computing: chance to beat the top machines in the Green500 supercomputer list

World's Most Efficient Dataflow Supercomputer at STFC Daresbury Laboratory to Drive UK Science and Innovation

The Science and Technology Facilities Council (STFC) and Maxeler are collaborating in a project funded by the UK Department of Business Innovation and Skills to install the next generation of supercomputing technology in a new facility at the Daresbury Laboratory focusing on energy efficient computing research. The supercomputer will offer orders of magnitude improvement in performance and efficiency. The new MPC-X supercomputer will be available in Summer 2014 and will allow UK industry and academia to develop products and services based on MPC data analytics engines for applications domains such as medical imaging and healthcare data analytics, manufacturing, industrial microscopy, large scale simulations, security, real-time operations risk, and media/entertainment.

The dataflow supercomputer will feature Maxeler developed MPC-X nodes capable of an equivalent 8.52TFLOPs per 1U and 8.97 GFLOPs/Watt, a performance per Watt that tops the Green500 today. MPC-X nodes build on the previous generation technology from Maxeler deployed at JP Morgan where real-time risk computation equivalent to 12000 x86 cores was achieved in 40U of dataflow engines. For the full story please visit our website at:

http://www.maxeler.com/stfc-dataflow-supercomputer/?utm_source=Commercial+List&utm_campaign=862e1b9d0e-CommercialFebruary2014_Mailer&utm_medium=email&utm_term=0_ece0f8fd2e-862e1b9d0e-336335821.

MAXELER Technologies

Search

Platforms Products Technology About Us MyMaxeler

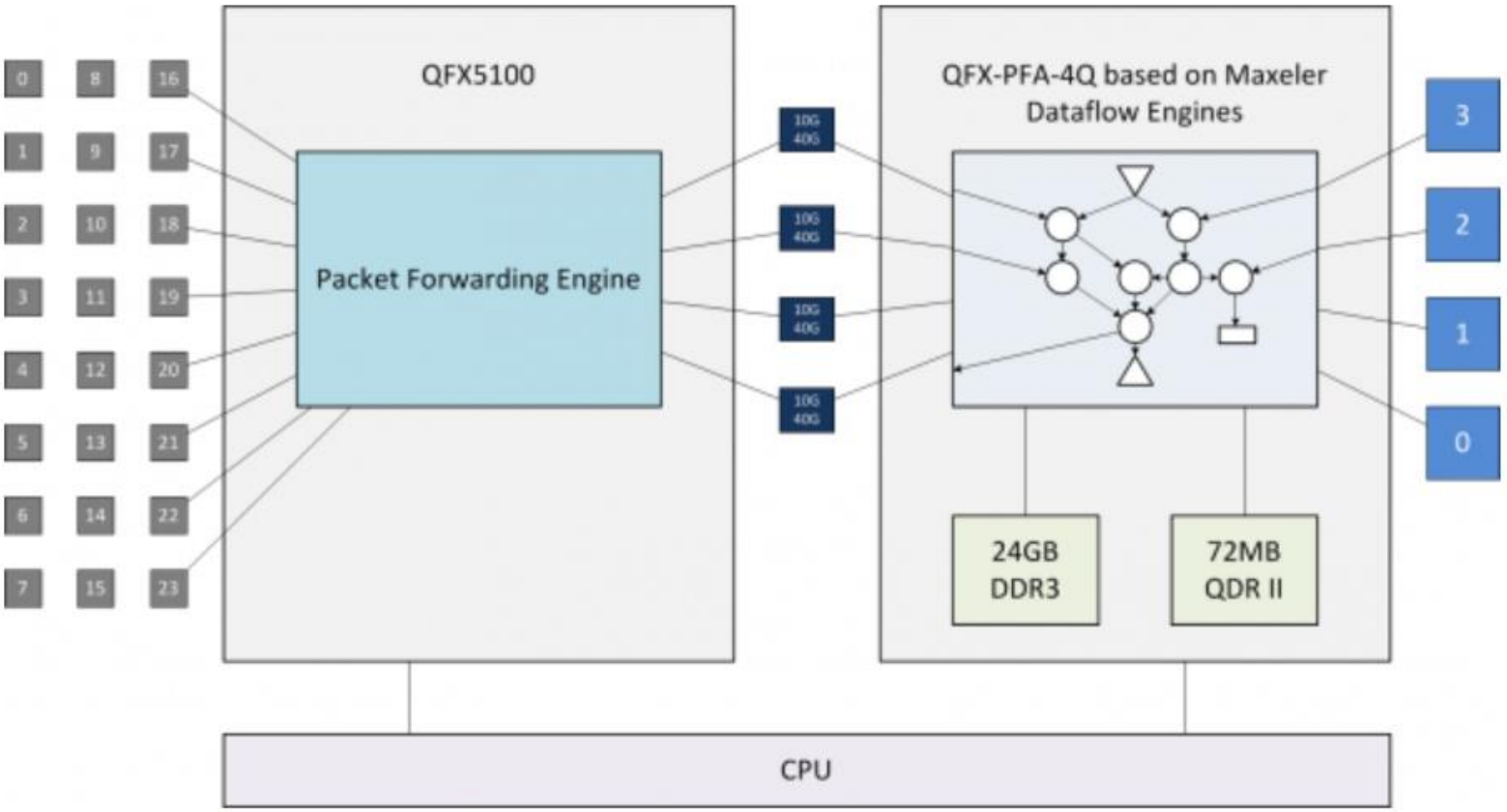
MPC-X Series MPC-C Series MPC-N Series JDFE (QFX-PFA-4Q) Desktop MaxCloud Software Training

JDFE

Juniper QFX5100 with QFX-PFA-4Q

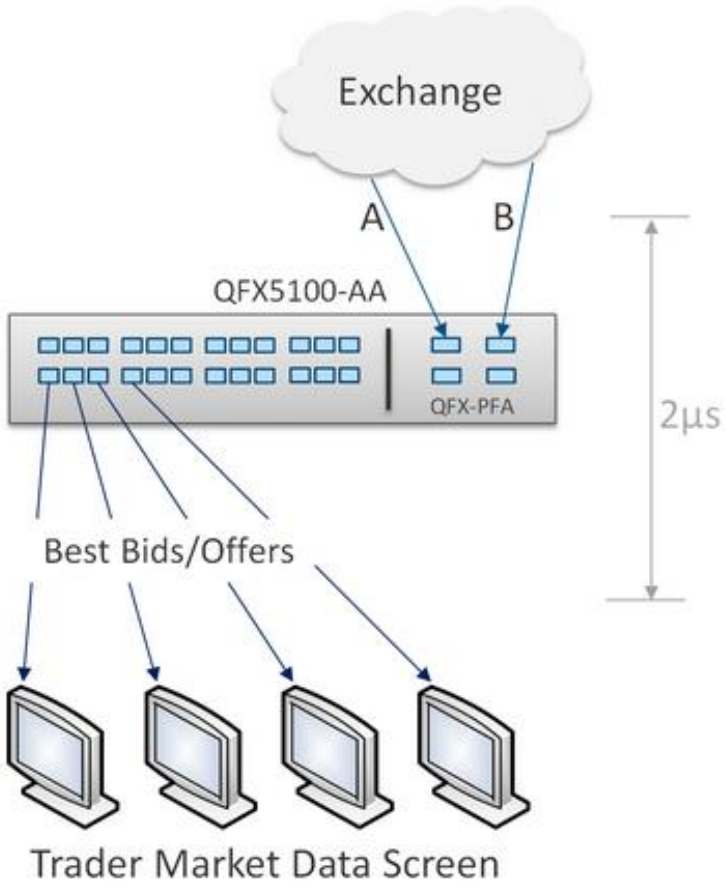
QFX-PFA-4Q based on Maxeler Dataflow Engines

Juniper: System Design (www.maxeler.com.products/jdfe)

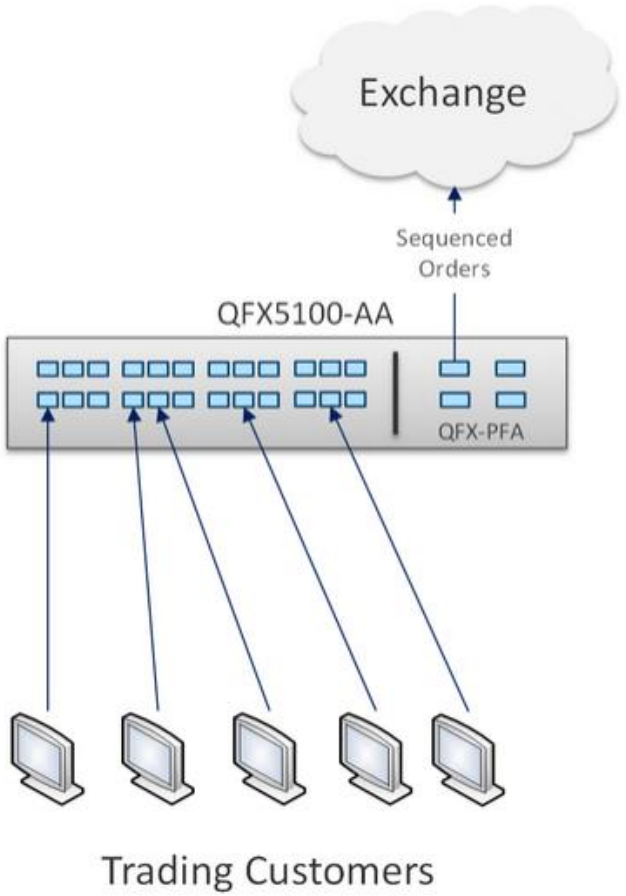


Juniper: Example Use Cases

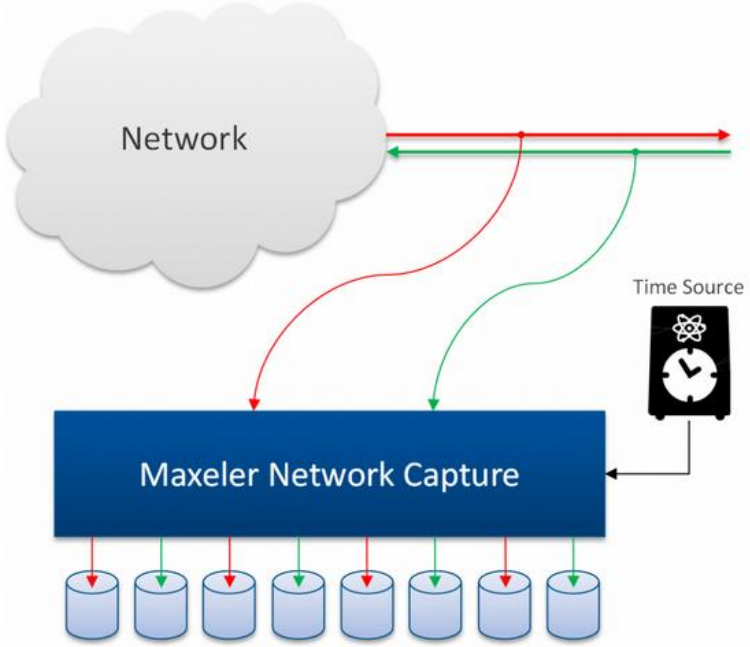
Exchange Market Data Dispetcher



FIFO Order Gateway



Network Packet Capture



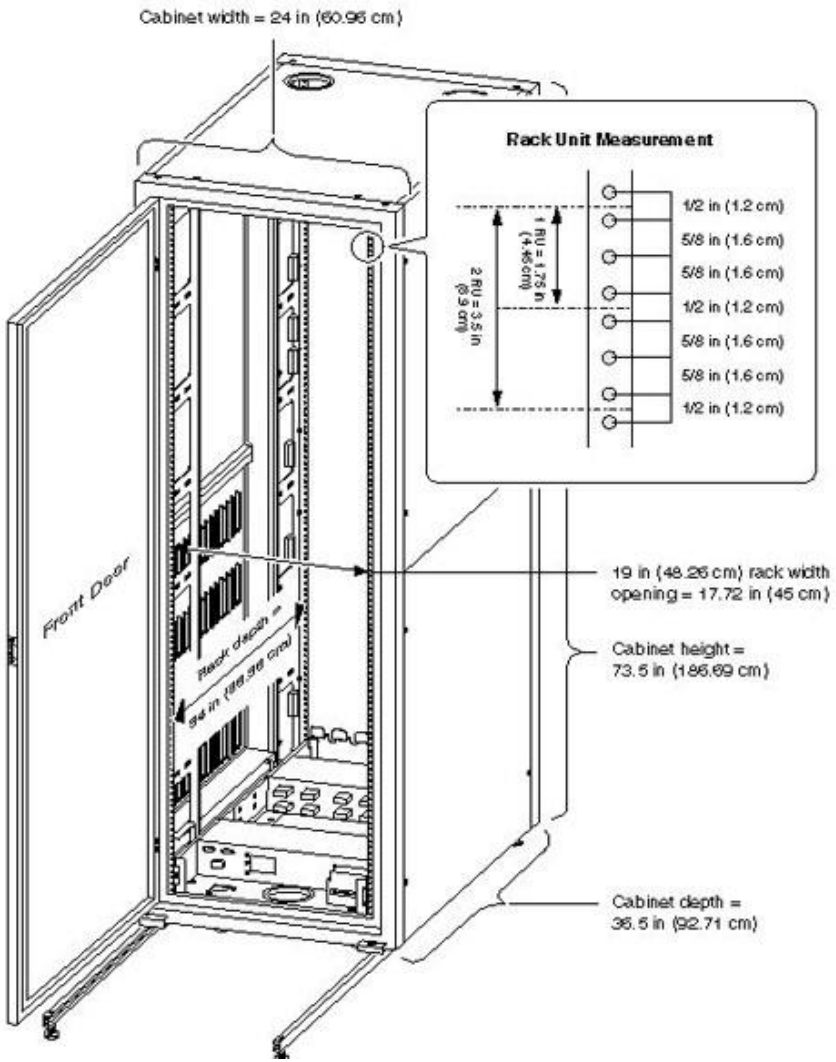
- Provides sustained line-rate packet capture in distributed write mode and at bursts of up to 24GB in size in local write mode.
- Custom on-chip application layer (DPI) filtering and compression can be used to further reduce storage overhead and consolidate relevant data.
- The application configures pairs of DFE SFP ports into ‘tapped’ pass through connections. Ethernet data is captured from these ‘tapped’ connections and written out to pcap files.
- The DFE implementation sustains more reliable throughput than commodity software based packet capture solutions including deep packet filtering, distributed writing, and a large DRAM buffer.

Operational Costs – OPEX

- Size matters:
  - Real estate is expensive
  - Managing and operating real estate is expensive
  - Portability matters for many applications
  - Communication delay depends on distance from one corner to the other…smaller is faster.

- Electricity matters:
  - Over 50% of cost of computing (disregarding SysAdmin and programmer salaries) are in electricity.

What is a Rack, what is 1U?



A rack holds computing units of 1U, 2U, 3U, 4U

1U = 19inch × 36.5inch × **1.75inch**
2U = 19inch × 36.5inch × **3.5inch**

Each compute unit has it's own power supply

1U CPU servers can have 1-2 mother boards
each with multiple CPU chips
each with multiple cores.

1U DFE boxes can have 8 DFEs or
[6 DFEs + 2 CPU chips], connected via InfiniBand

Each 1U DFE box has 384GB of DRAM
and 48 independent memory channels.

Cost of Programming

When writing a line of code,
can you tell how much operational costs you create
by deciding to write a program in one way
versus another?

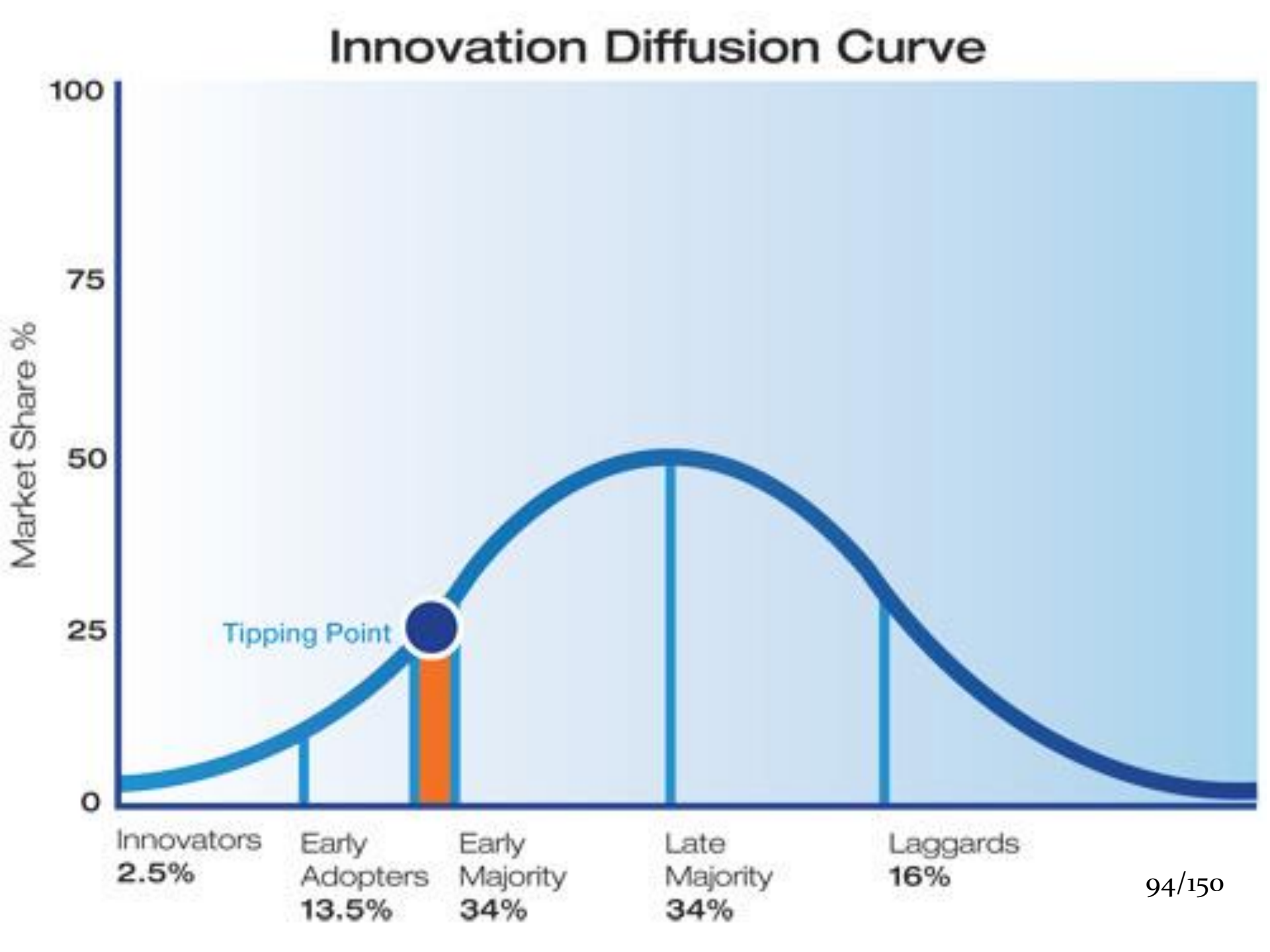
The DFE programming is OPEX aware programming!

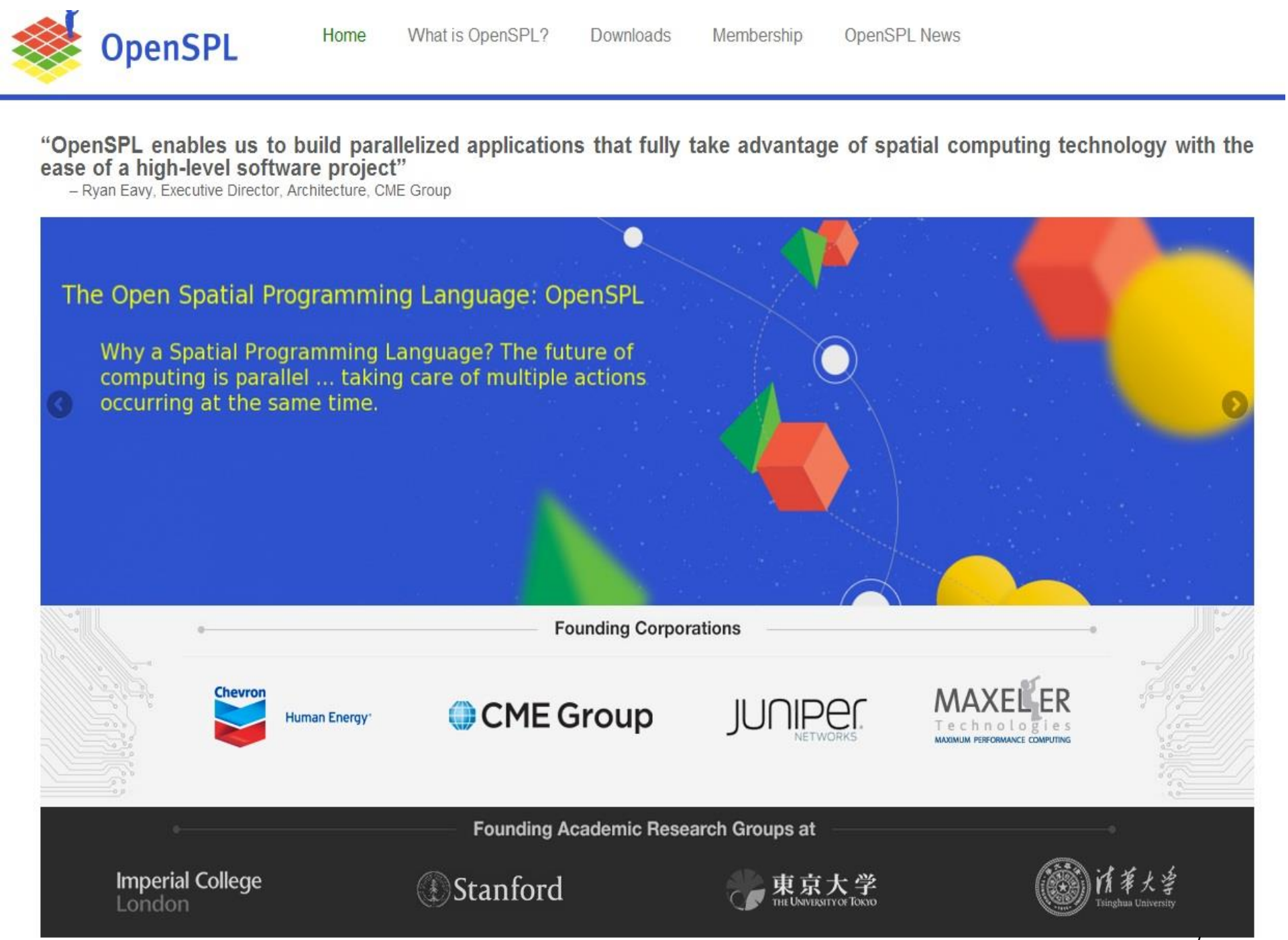
A programmer
(and his boss watching the project deadline)
values his own time right now
more than recurring future OPEX costs.

MAX-UP Members

The Maxeler University Program has over a hundred members across the world. Check out the map below for the latest membership.

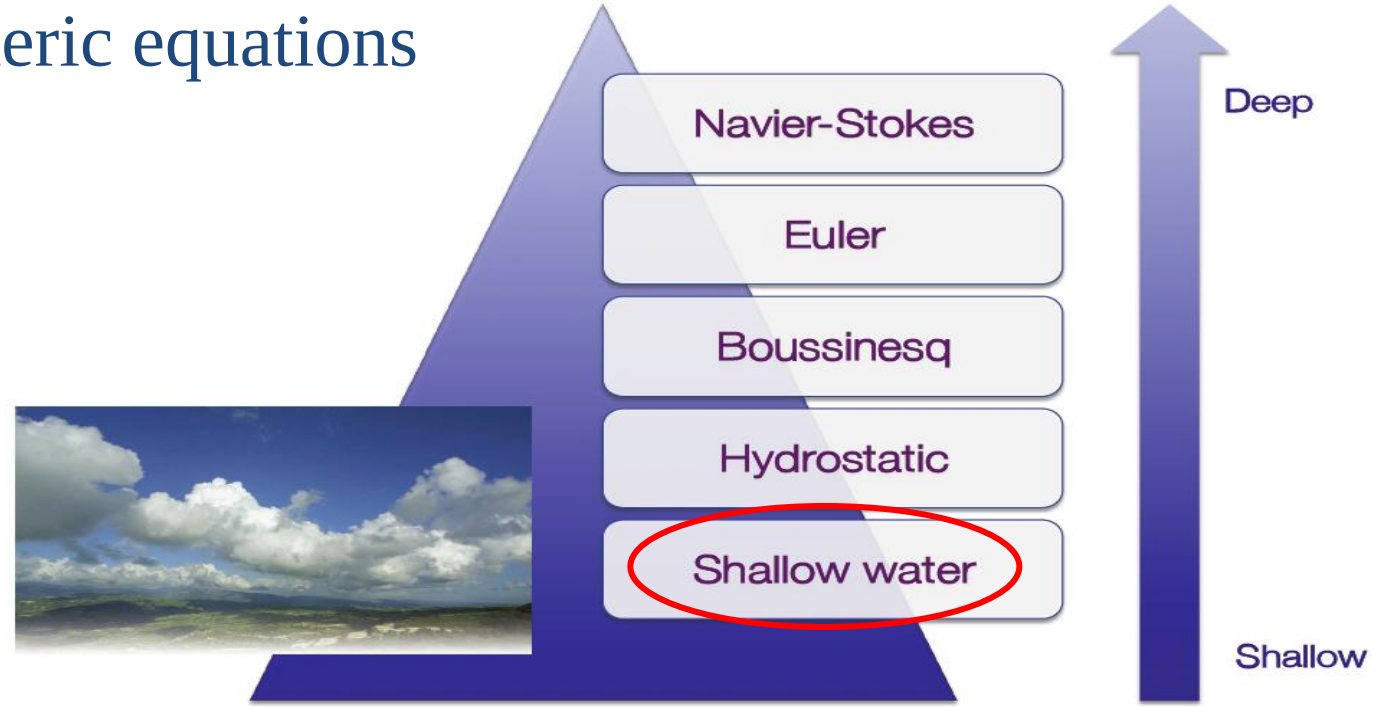
1.1.2014.


Innovation Diffusion Curve
100
75
50
25
0
Market Share %
Tipping Point
Innovators
2.5%
Early
Adopters
13.5%
Early
Majority
34%
Late
Majority
34%
Laggards
16%


OpenSPL
Home
What is OpenSPL?
Downloads
Membership
OpenSPL News
"OpenSPL enables us to build parallelized applications that fully take advantage of spatial computing technology with the ease of a high-level software project"
– Ryan Eavy, Executive Director, Architecture, CME Group
The Open Spatial Programming Language: OpenSPL
Why a Spatial Programming Language? The future of computing is parallel ... taking care of multiple actions occurring at the same time.
Founding Corporations
Chevron
Human Energy
CME Group
JUNIPER NETWORKS
MAXELER Technologies
MAXIMUM PERFORMANCE COMPUTING
Founding Academic Research Groups at
Imperial College London
Stanford
東京大学
THE UNIVERSITY OF TOKYO
清華大學
Tsinghua University

Global Weather Simulation

- Atmospheric equations



- Equations: Shallow Water Equations (SWEs)

$$\frac{\partial Q}{\partial t} + \frac{1}{\Lambda}\frac{\partial(\Lambda F^1)}{\partial x^1} + \frac{1}{\Lambda}\frac{\partial(\Lambda F^1)}{\partial x^2} + S = 0$$

[L. Gan, H. Fu, W. Luk, C. Yang, W. Xue, X. Huang, Y. Zhang, and G. Yang, Accelerating solvers for global atmospheric equations through mixed-precision data flow engine, FPL2013]

Weather Model – Performance Gain

Platform	Performance ()	Speedup
6-core CPU	4.66K	1
Tianhe-1A node	110.38K	23x
MaxWorkstation	468.1K	**100x**
MaxNode	1.54M	**330x**

14x

Meshsize: 1024 × 1024 × 6

MaxNode speedup over Tianhe node: 14 times

Imperial College London

ISCAS

Weather Model -- Power Efficiency

Platform	Efficiency ( )	Speedup
6-core CPU	20.71	1
Tianhe-1A node	306.6	14.8x
MaxWorkstation	2.52K	**121.6x**
MaxNode	3K	**144.9x**

Meshsize: 1024 × 1024 × 6

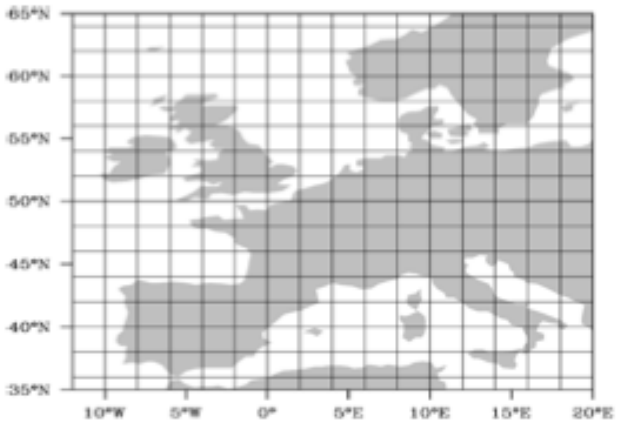
9 x

MaxNode is 9 times more power efficient

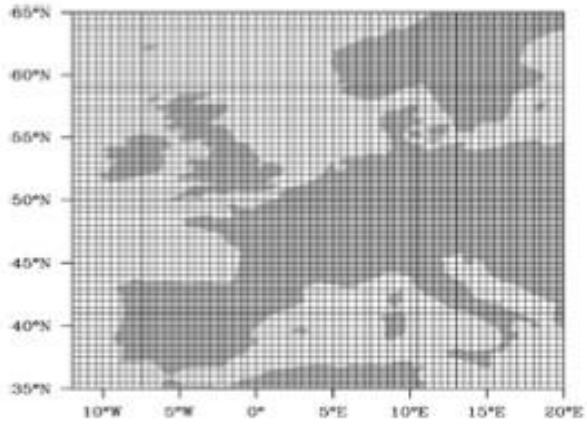
Imperial College London

ISCAS

Weather and Climate Models



Which one is better?



Finer grid and higher precision are obviously preferred but the computational requirements will increase ➔ Power usage → $$

What about using reduced precision? (15 bits instead of 64 double precision FP)

We use only **15 bits** for 98% of the computation:


MAXELER
Technologies
MAXIMUM PERFORMANCE COMPUTING
HPC
Low latency
Cloud
Computin
Seismic


MAXE
Techno
spective

https://www.maxeler.com

MAXELER Technologies

Platforms Products Technology About Us

Search

MyMaxeler

Maximum Performance Computer | Maxeler wins Frost & Sullivan Best Practices Award for Dataflow Computing | Maxeler featured in Wall Street Journal | Trading and Risk: What's Your Favorite Risk Flavor?

Maximum Performance Computing

Maxeler develops and delivers innovative High Performance Computing solutions

We combine Maximum Performance Computing consulting with our dataflow computing technology to drive faster deployment of next-generation algorithms and lasting competitive advantage.

Oil & Gas

Accelerated processing of large seismic datasets »

Financial Analytics

Compute value and risk in real time »

Low Latency

Respond to financial market events in nanoseconds »

Scientific Computing

30x reduced power consumption for supercomputers »

Latest News

March 13th 2014

Flying the Flag for UK Tech Innovation in Europe »

More news

February 25th 2014

STFC Daresbury Laboratory first to install Maximum Performance Computer (MPC) »

Our Publications

May 5th 2013

Moving from Petaflops to Petadata »

Contact us »

LinkedIn

Follow us

Careers

https://www.maxeler.com/moving-from-petaflops-to-petadata/

MAXELER Technologies

Platforms Products Technology About Us

Search

MyMaxeler

Back...

Moving from Petaflops to Petadata

May 5, 2013

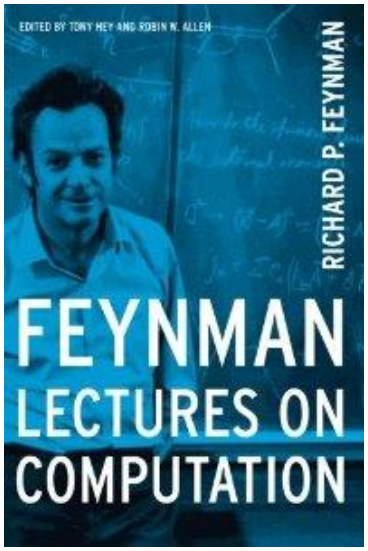
VIEWPOINT: Moving from Petaflops to Petadata

M. Flynn[‡‖], O. Mencer[‡∓], V. Milutinovic[†], G. Rakocevic[§], P. Stenstrom[ϑ], R.Trobec[þ], M. Valero[F]

[‡]Maxeler Technologies, [‖]Stanford University, [∓] Imperial College London, [†]University of Belgrade, [§]Mathematical Institute of the Serbian Academy of Sciences and Arts in Belgrade, [ϑ] Chalmers University of Technology, [þ]Jožef Stefan Institute, [F]Barcelona Supercomputing Centre

Communications of the ACM, Vol. 56 No. 5 May 2013, doi: 10.1145/2447976.2447989

Richard Feynman on Computation



In theory, a computer system
can be constructed which uses **NO ENERGY**.

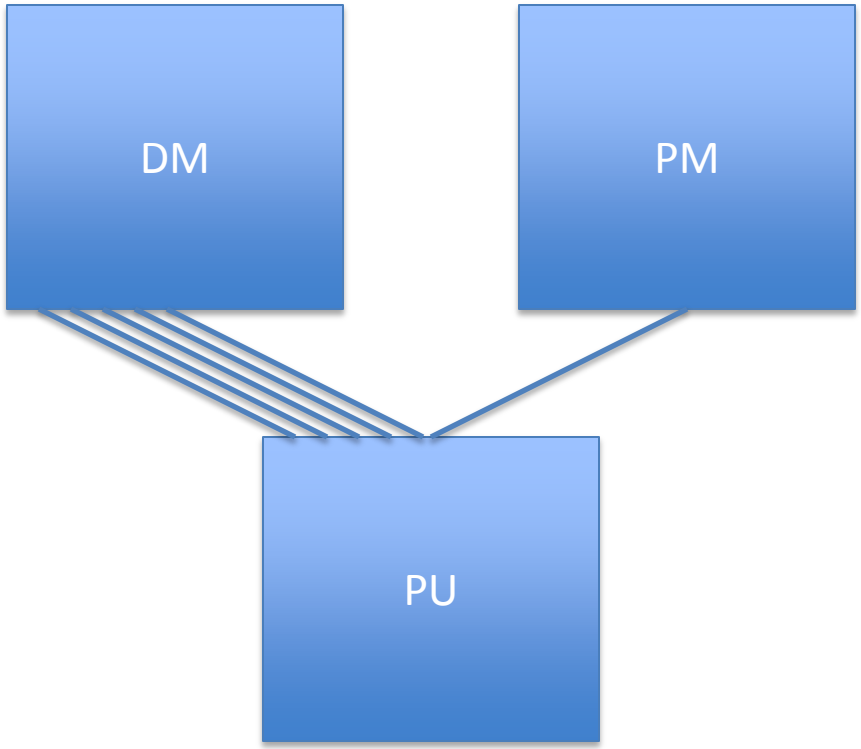
Energy is only needed when **information** is lost.

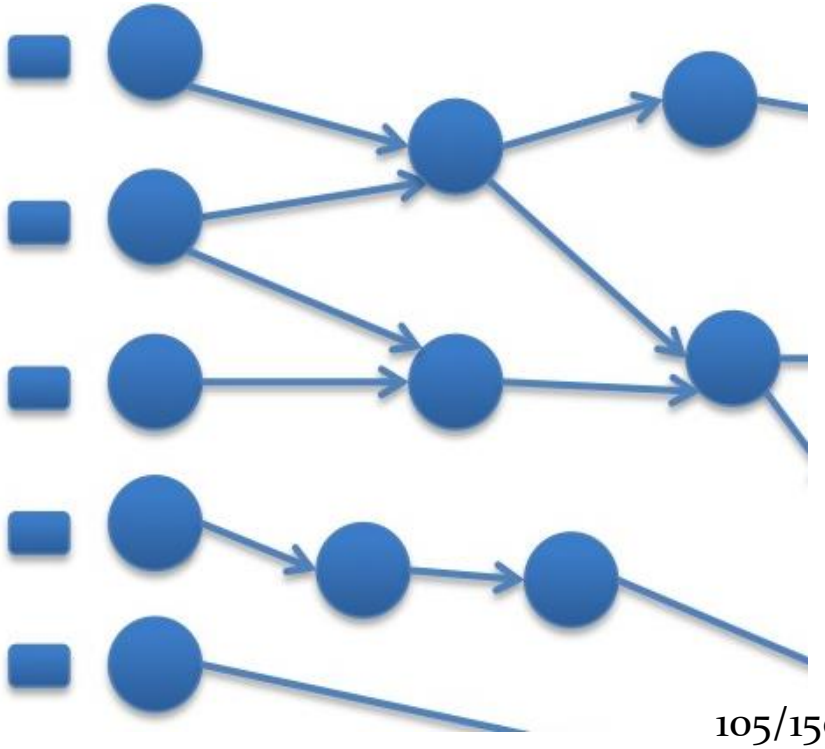
Reordering of **information** does not require energy
from a pure physics perspective.

Of course, **moving information** takes energy…

Essence: Feynman

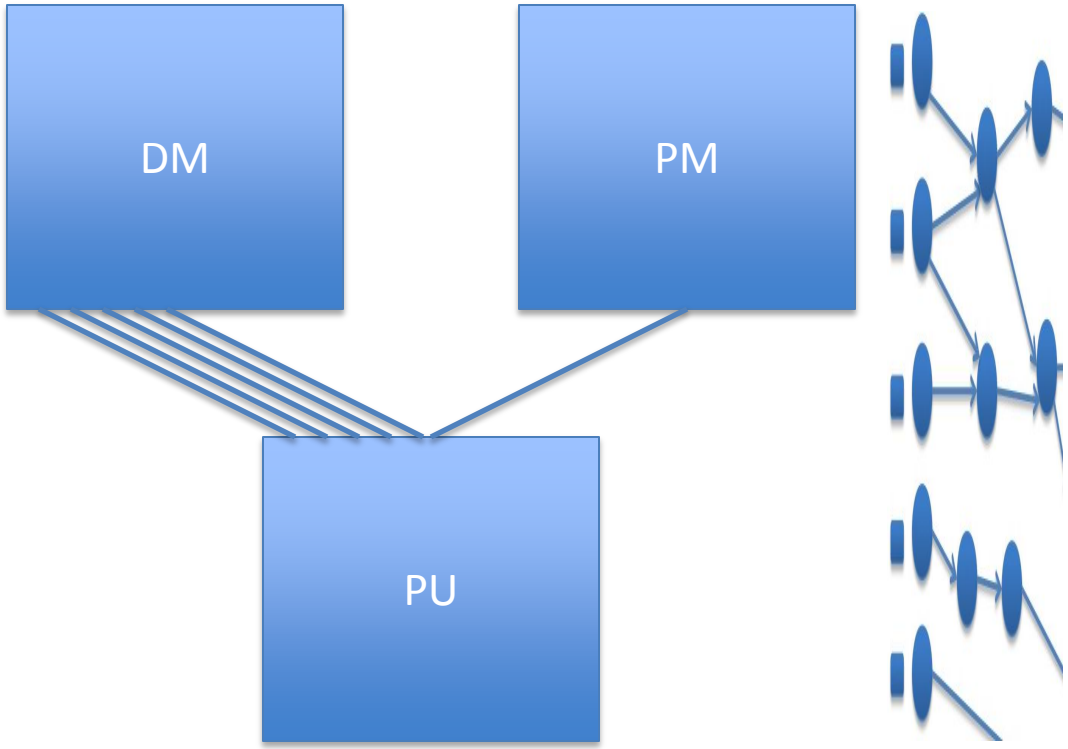
- ALU possible at zero power (Arithmetic+Logic)
- COMM not possible at zero power (MEM+MPS)





Essence: Feynman

- ALU possible at zero power (Arithmetic+Logic)
- COMM not possible at zero power (MEM+MPS)



~~The Power Challenge~~
The Data Movement Challenge

	Today	2018-20
Double precision FLOP	100pj	10pj

- Moving data on-chip will use 12x more energy than computing
- Moving data off-chip will use **200x more energy than computing**!
- New optimization criteria on the rise!
  - Before: Minimizing the number of steps and the size of each step (MLTP.min)
  - Now: Minimizing the length of exe.graph.edges and making the execution graph mappable.

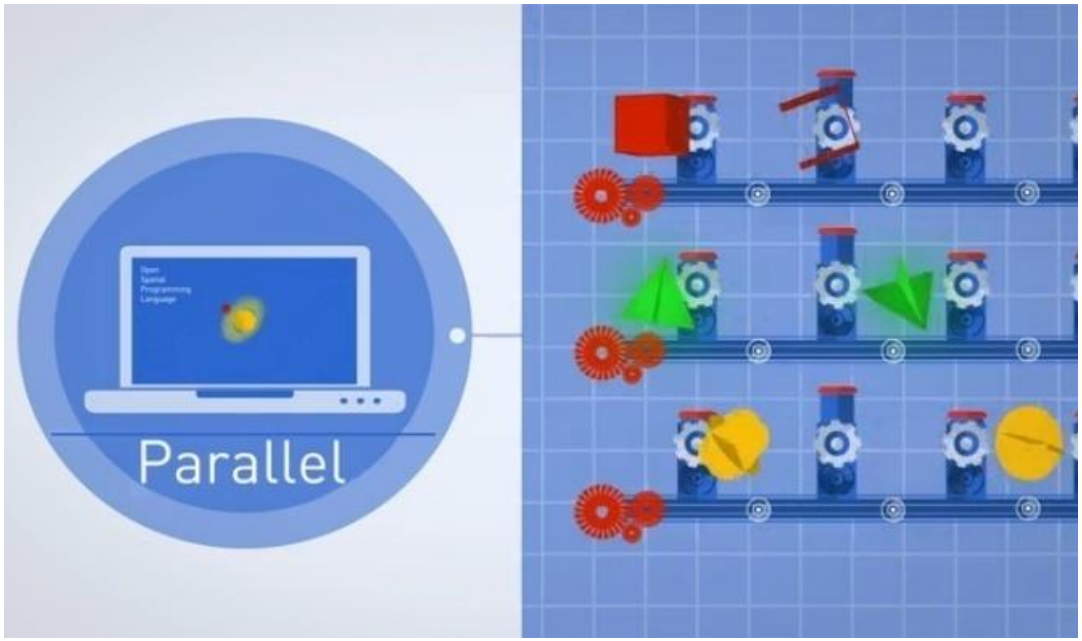
A new way of thinking about

FPGA OpenSPL

Suggesting a look at Mike Flynn's narration: www.openspl.org

Overview

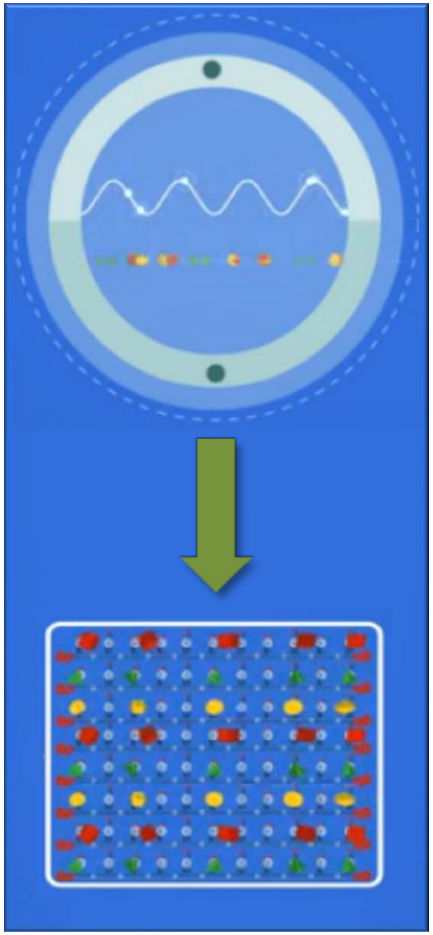
- ❑What is OpenSPL?
- ❑Motivation for spatial programming
- ❑Transforming the mindset…
- ❑Controlflow & Dataflow
- ❑Basics of OpenSPL
- ❑Challenges
- ❑Examples
- ❑Summary and Roadmap



What is OpenSPL?

Open Spatial Programming Language

- Allows programs to operate more effectively and efficiently by utilizing the space rather than depending only on time
- Embrace the natural parallelism of the substrate
- Data is transformed as it flows through the fabric
- Improve computational density
- General purpose development semantics

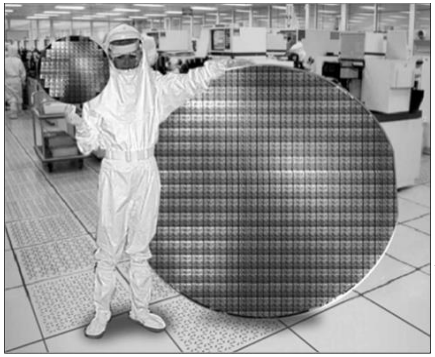


Motivation for Programming in Space

- Core clock frequencies evened out in the few GHz range
- Energy / Power consumption of modern HPC systems
  became huge economic burden not to be ignored any longer
- Specialization has proven its power efficiency potentials
- The requirements for annual performance improvements keep growing steadily
- SoCs are now exploiting also the third dimension (3D-int)
- However, the majority of programmers build upon the legacy,
  1D linear view and sequential execution
- Many clever proposals but no good solution to date
  (e.g., Cilk, Sequoia, OmpSs and OpenCL)

Moore Motivation…

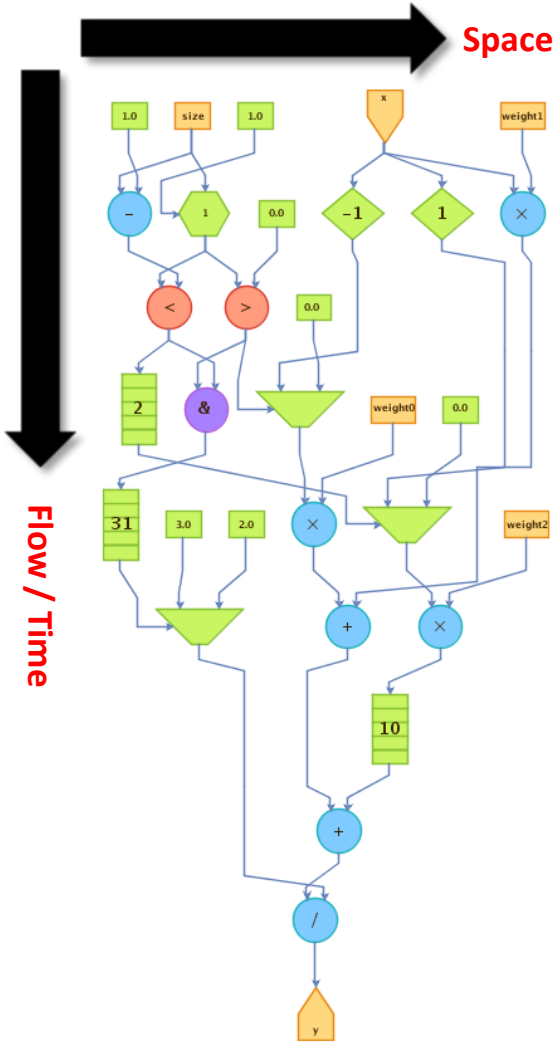
- The number of transistors on a chip keeps scaling
  - Between 2003 and 2013 it went up from 400M (Itanium 2) to 5 Bln (Xeon Phi) in the case of modern processors
- Exploding data volumes while memory can't follow
  - In the same period DRAM latency improved by less than 3x



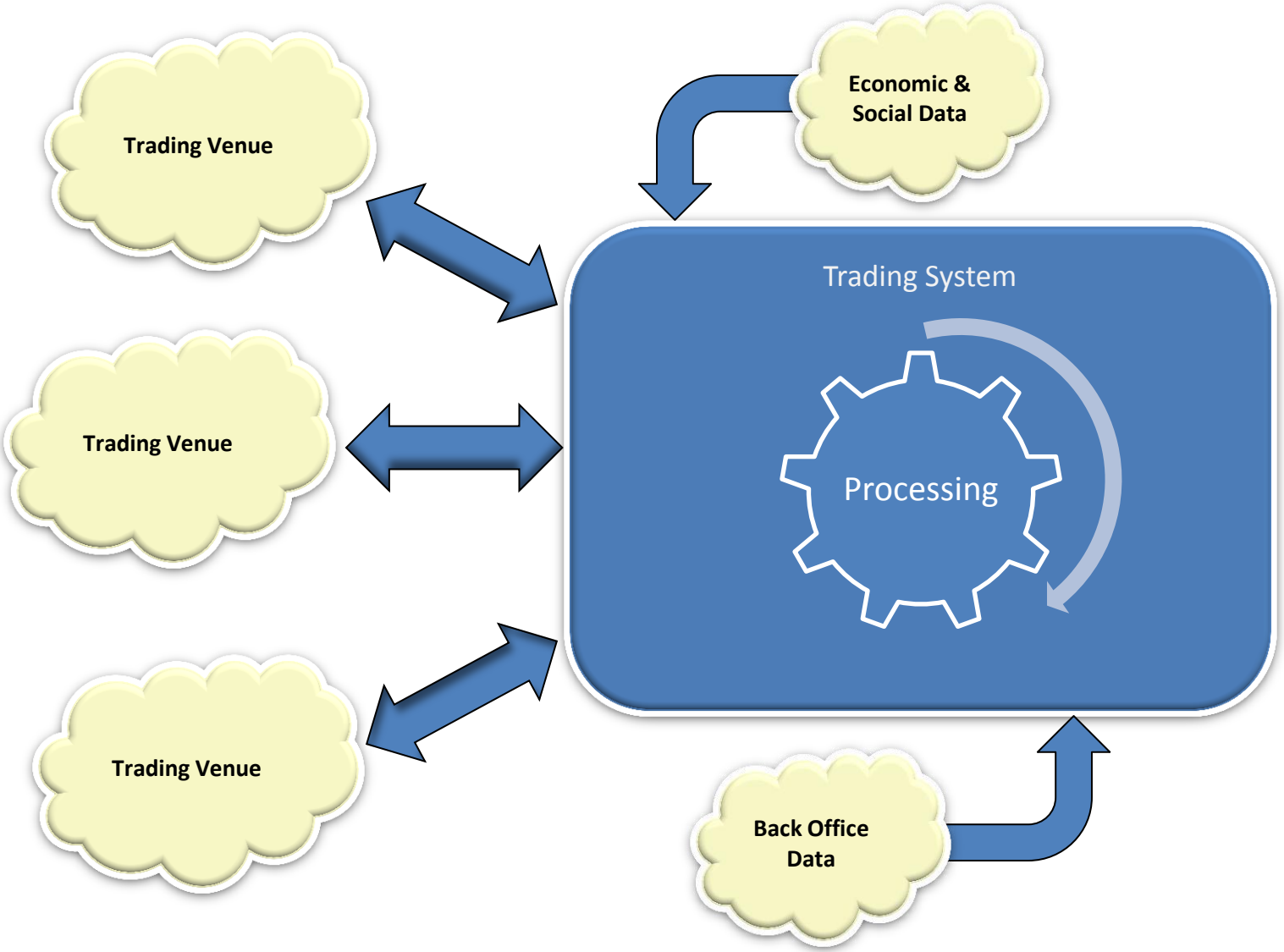
One's "dream" about more of Moore (courtesy of Intel)

Transforming the Mindset …

- Thinking in space rather than in time
- Difficult change in mindset to overcome
- Transformation of data through flow over time
- Instructions are parallelized across the available space

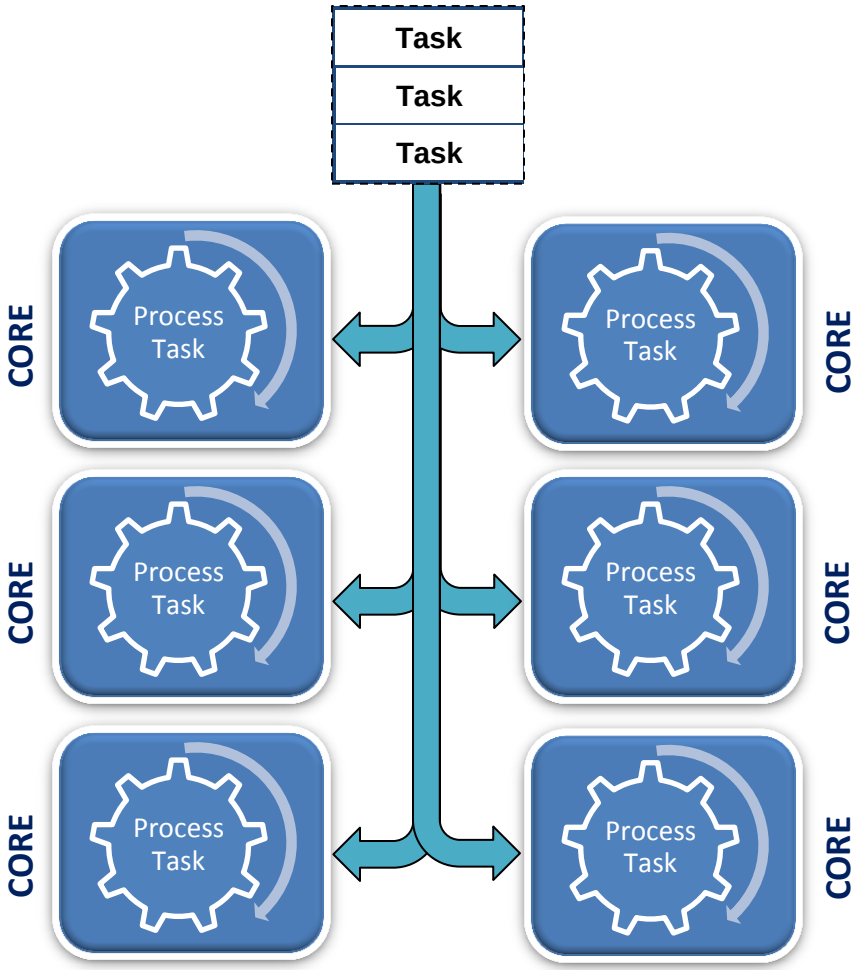


Flow of data (Financial Example)



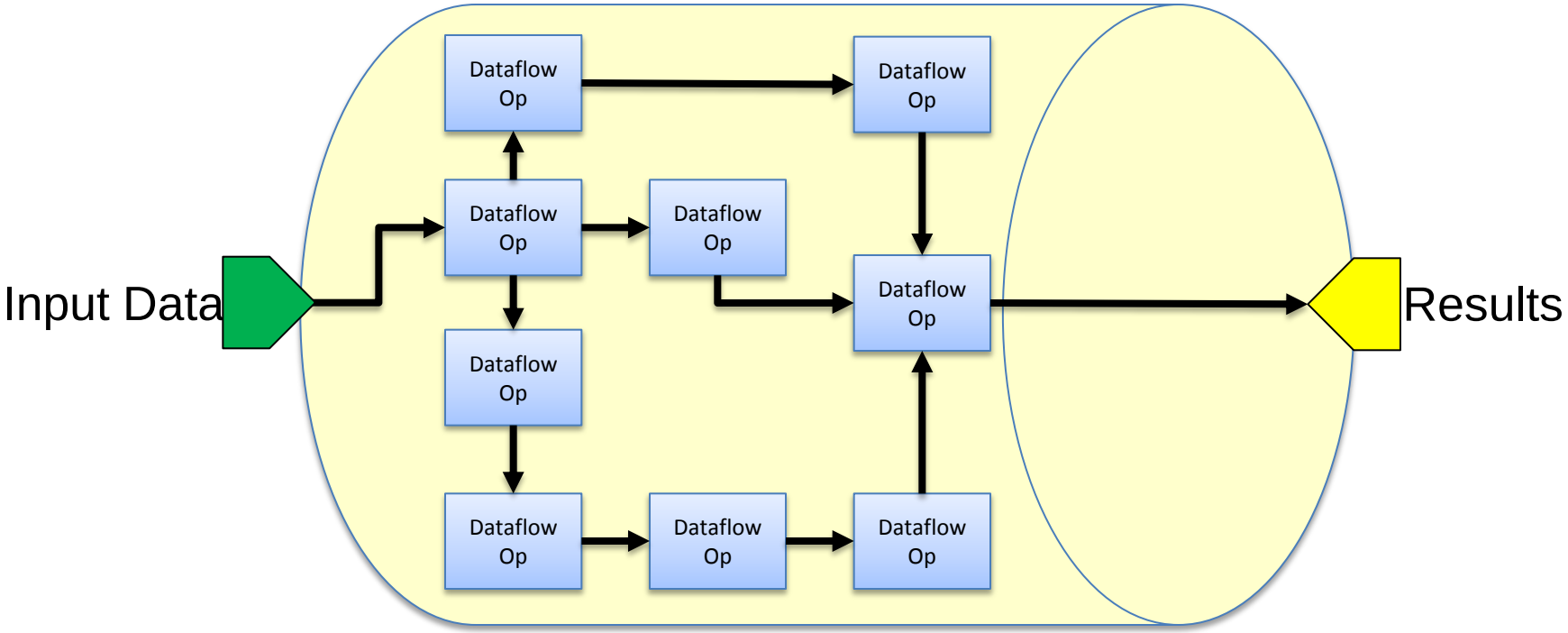
Parallelism and Control Flow

- Traditional control flow
- Pragmas to enable multitasking
- Tasks distributed among many time-slices of CPU cores



Parallelism and Data Flow

- Program instructions are laid down on the chip surface in 2D
- All instructions (Dataflow Op) execute in parallel on every clock tick
- Results are immediately used by the next Op



OpenSPL Basics

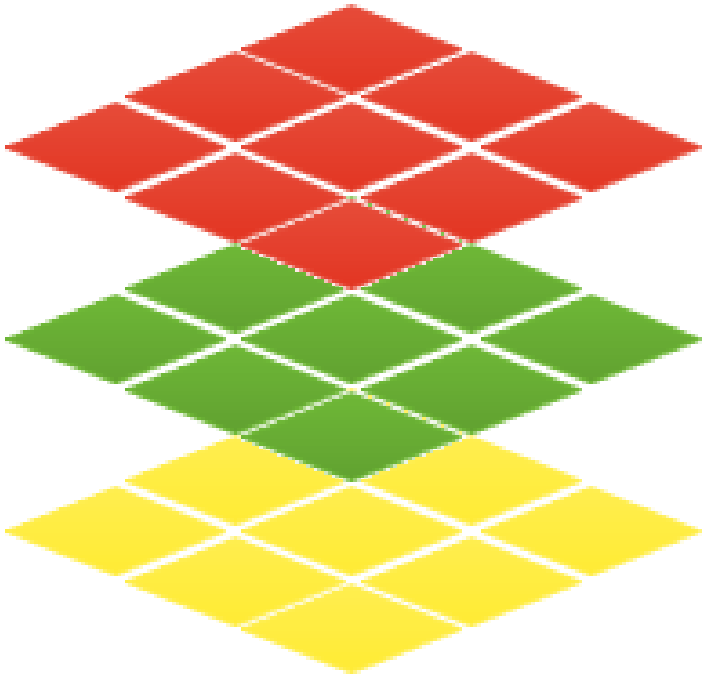
- Controlflow and Dataflow are decoupled
  - Both are fully programmable
- Operations exist in space and by default run in parallel
  - Their number is limited only by the available space
- All operations can be customized at various levels
  - e.g., from algorithm down to the number representation
- Multiple operations constitute kernels
- Data streams through the operations / kernels
- Data transport and compute can be balanced
- All resources work all of the time for max performance
- In/Out data rates determine the operating frequency

Challenges with Native HDL

- Tools built for digital designers
- Description for a digital system
- Can't we think of a system as one or many streams of data transformation?
- Designed for describing **hardware,** not describing **computation**

```
entity NAME_OF_ENTITY is [ generic
generic_declarations);]
          port (signal_names: mode type;
                     signal_names: mode type;
                                :
                     signal_names: mode type);
end [NAME_OF_ENTITY] ;

architecture architecture_name of
NAME_OF_ENTITY is
-- Declarations
          -- components declarations
          -- signal declarations
          -- constant declarations
          -- function declarations
          -- procedure declarations
          -- type declarations

          :

begin
-- Statements

          :

end architecture_name;
```

Structure of OpenSPL

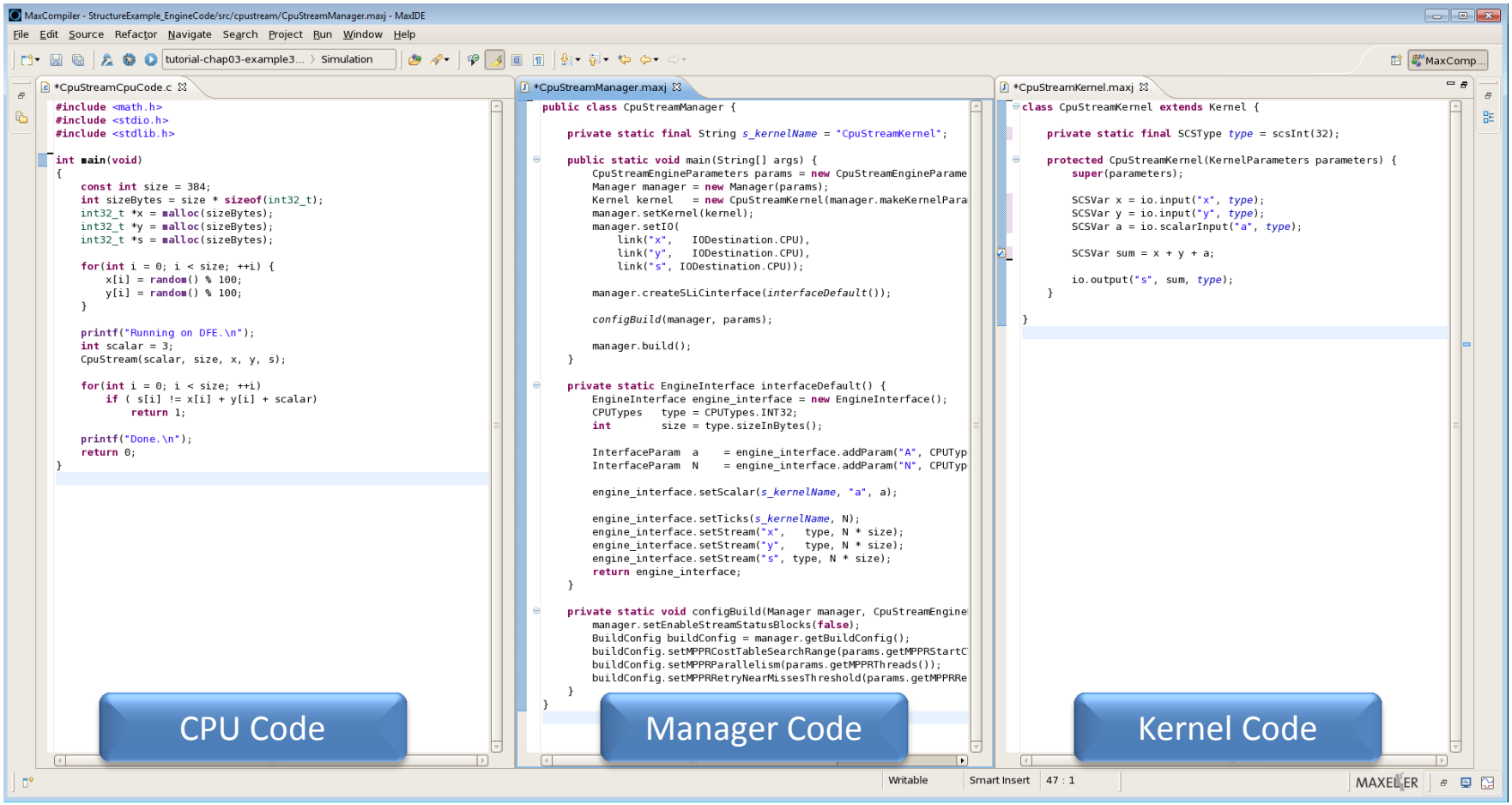


Host: Application setup, environment configuration, etc.

Manager: Substrate configuration, stream definitions, kernel configuration, etc.

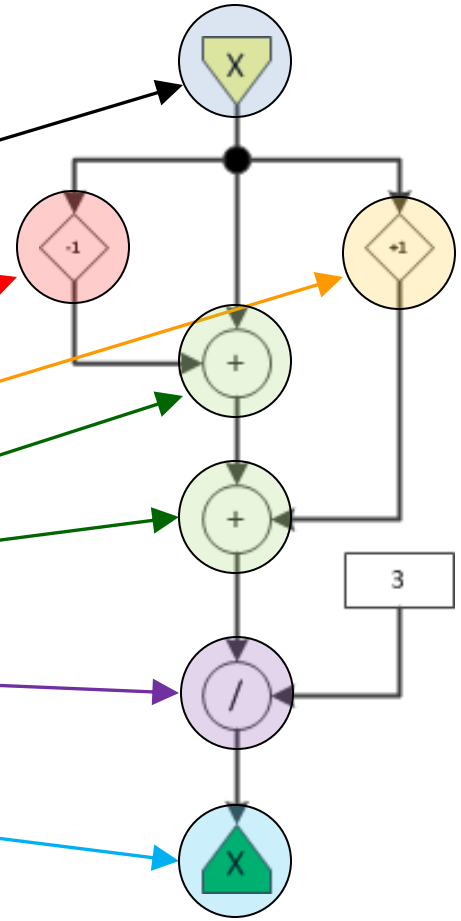
Kernel(s): Application Logic

Structure of OpenSPL – Example IDE



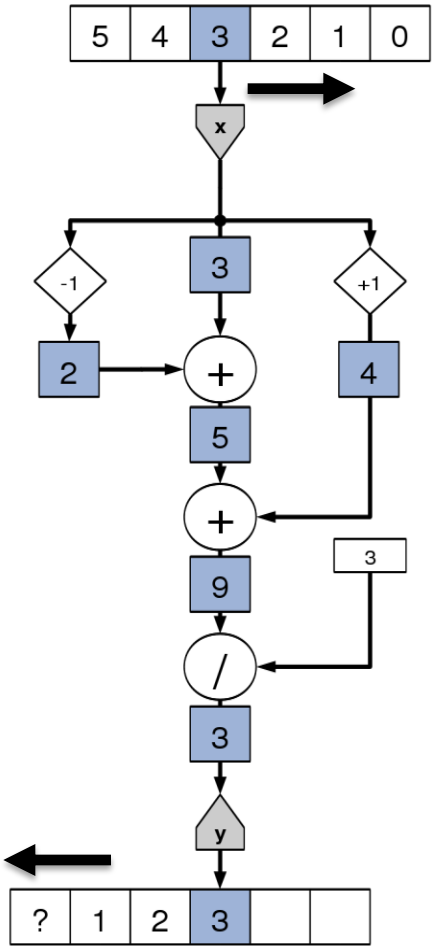
Example #1 – Simple Kernel

```
public class MovingAverageKernel extends Kernel {

    public MovingAverageKernel(KernelParameters parameters,
        int N) {
      super(parameters);

      //Input
      SCSVar x = io.input("x");

      //Data
      SCSVar prev = stream.offset(x, -1);

      SCSVar next = stream.offset(x, 1);

      SCSVar sum = prev+x+next;

      SCSVar result = sum/3;

      //Output
      io.output("y", result);
    }
}
```



Example #2 – Moving Average

```
class MovingAvgKernel extends Kernel {
    MovingAvgKernel() {
        SCSVar x = io.input("x");
        SCSVar prev = stream.offset(x, -1);
        SCSVar next = stream.offset(x,  1);
        SCSVar sum = prev + x + next;
        SCSVar result = sum / 3;
        io.output("y", result);
    }
}
```

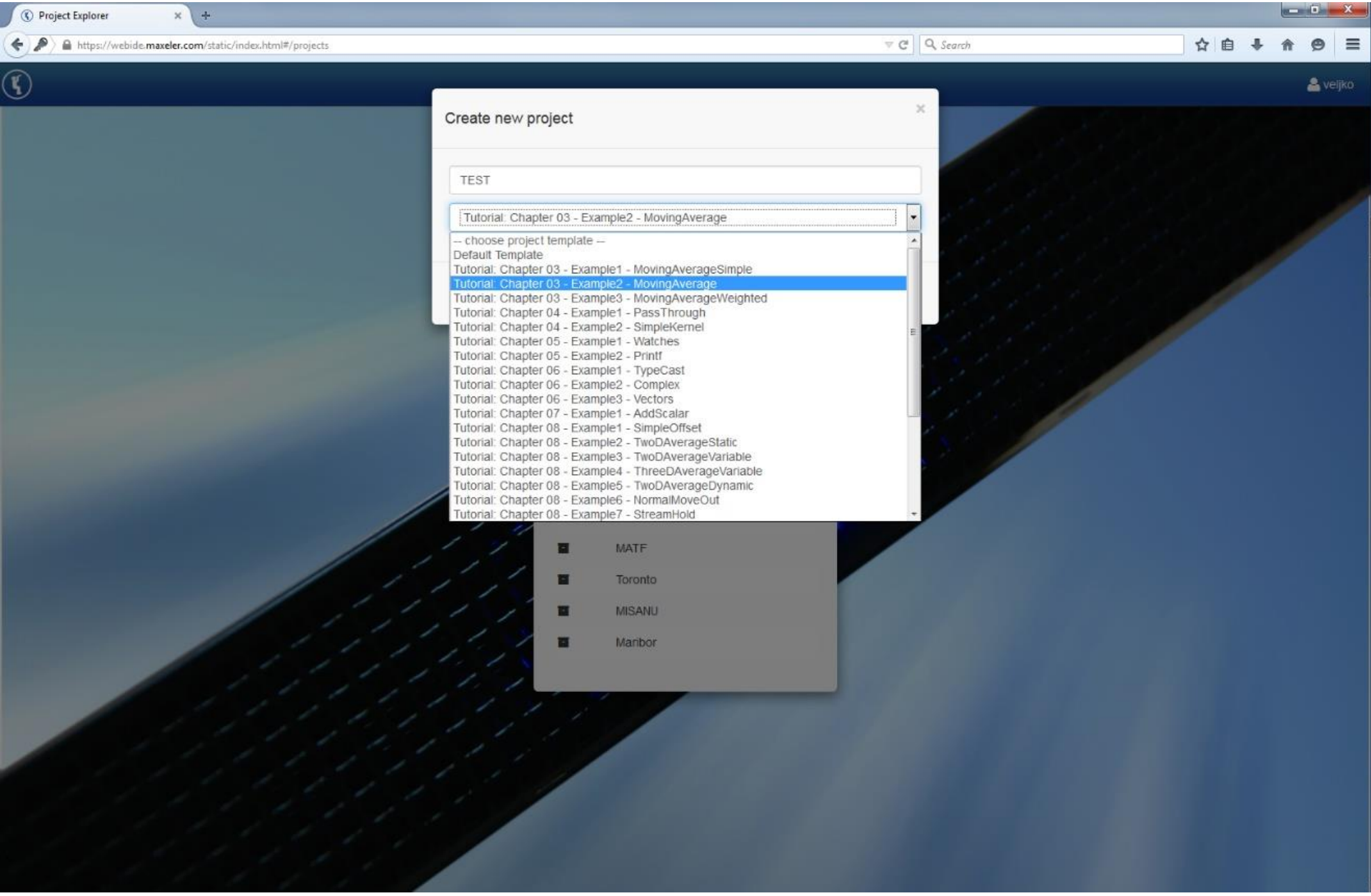


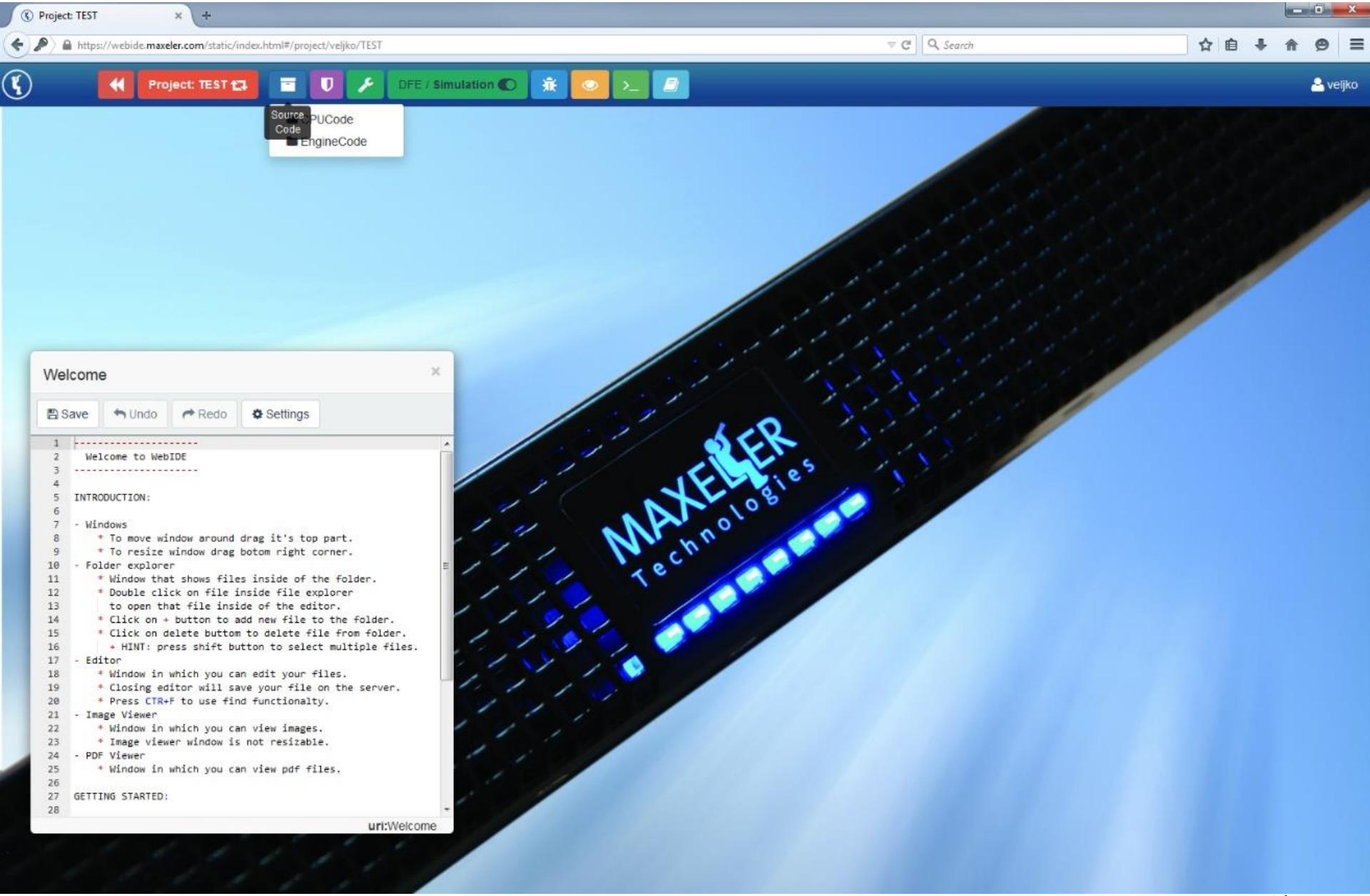
Example #3 – Implied Volatility

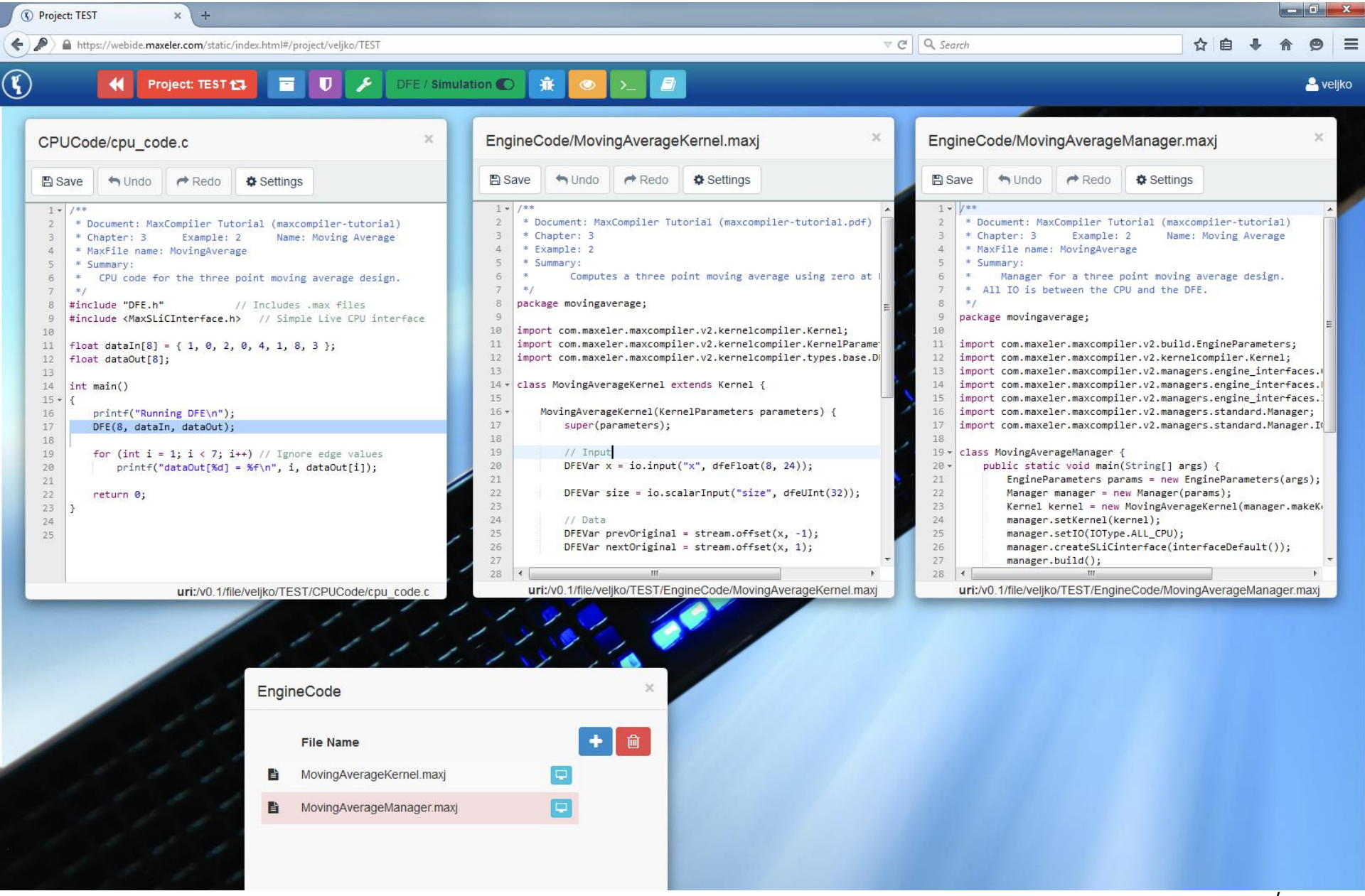
```
// Based on: "A New Formula for Computing Implied Volatility" by Steven Li
SCSVar impliedVol(SCSVar optionPrice,
                  SCSVar futurePrice,
                  SCSVar strikePrice,
                  SCSVar timeToExpiration,
                  SCSVar interestRate) {

    SCSVar discountFactor = exp(interestRate*timeToExpiration);

    optionPrice = optionPrice * discountFactor;

    SCSVar sqrtT = sqrt(timeToExpiration);

    SCSVar KmS = strikePrice - futurePrice;
    SCSVar SpK = futurePrice + strikePrice;

    SCSVar alpha = (sqrt(2.0*Math.PI) / SpK) * (optionPrice + optionPrice + KmS);

    SCSVar tempB = max(0, alpha*alpha - 4.0*KmS*KmS/(futurePrice*SpK));

    return 0.5*(alpha + sqrt(tempB)) / sqrtT;
}
```

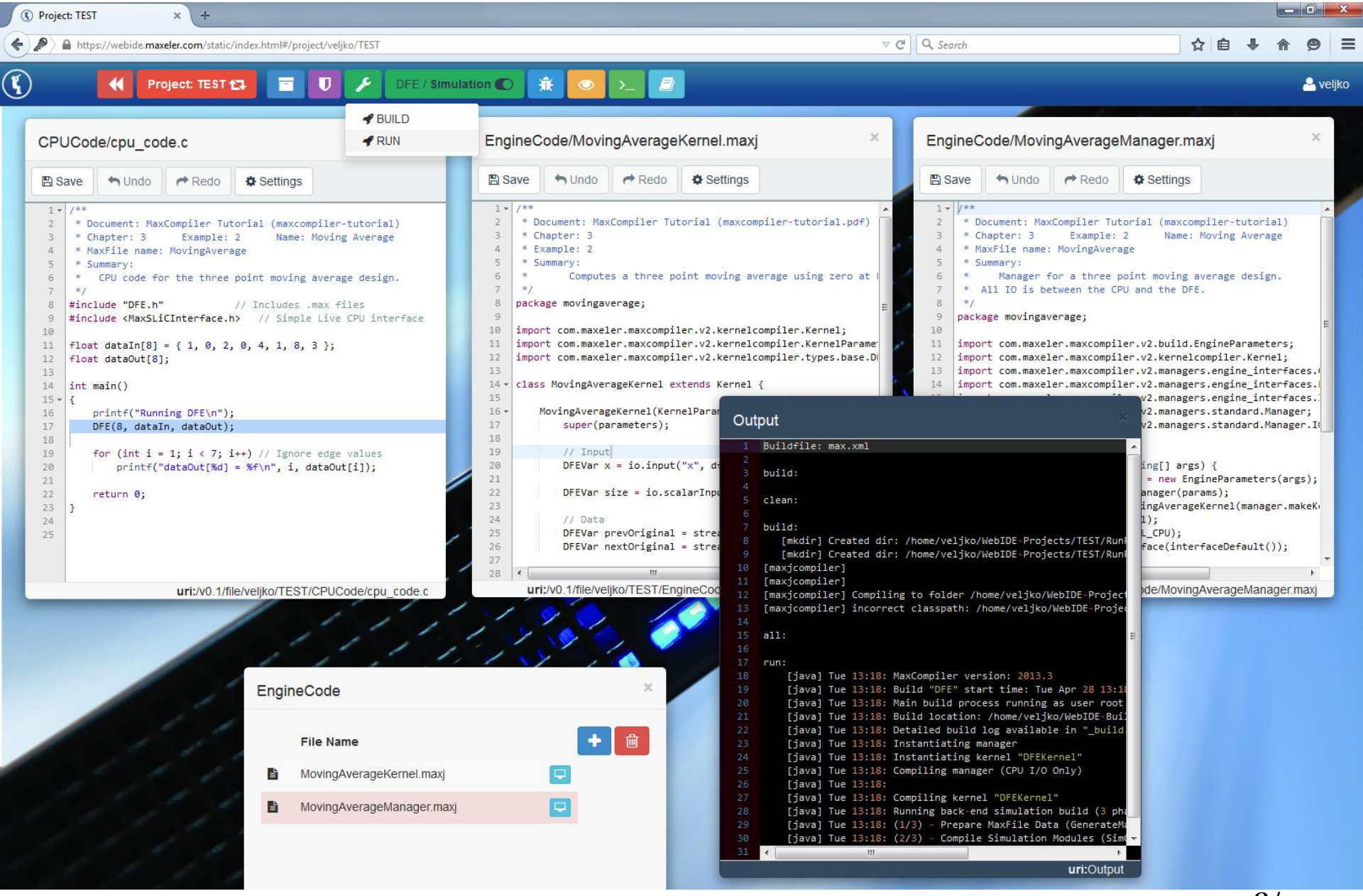
Running time: ~700ns

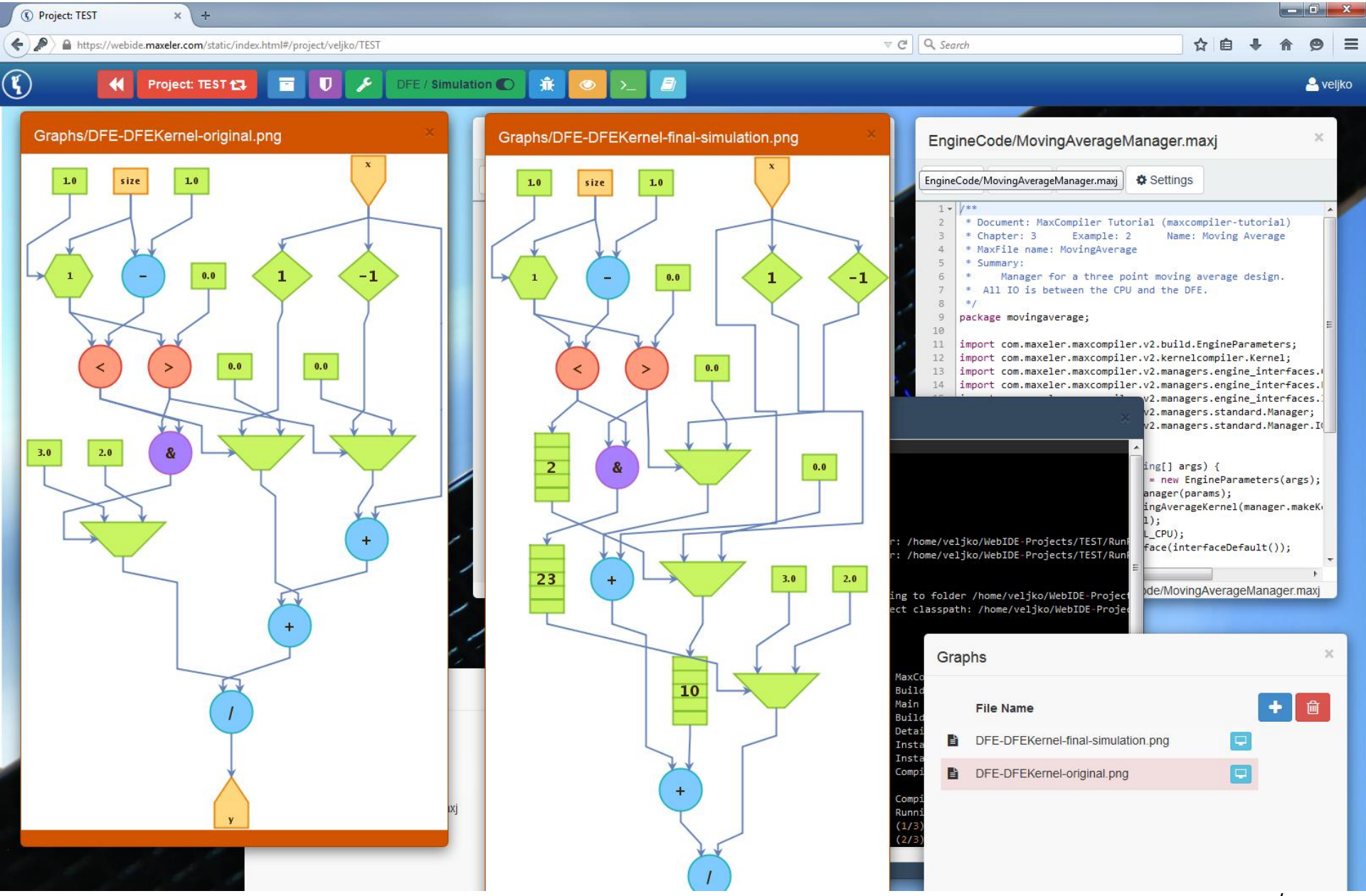

Login
https://webide.maxeler.com/static/index.html#/login
Search
Username
veljko
Password
Login
MAXELER
Technologies
webide.maxeler.com
maxeler.mi.sanu.ac.rs

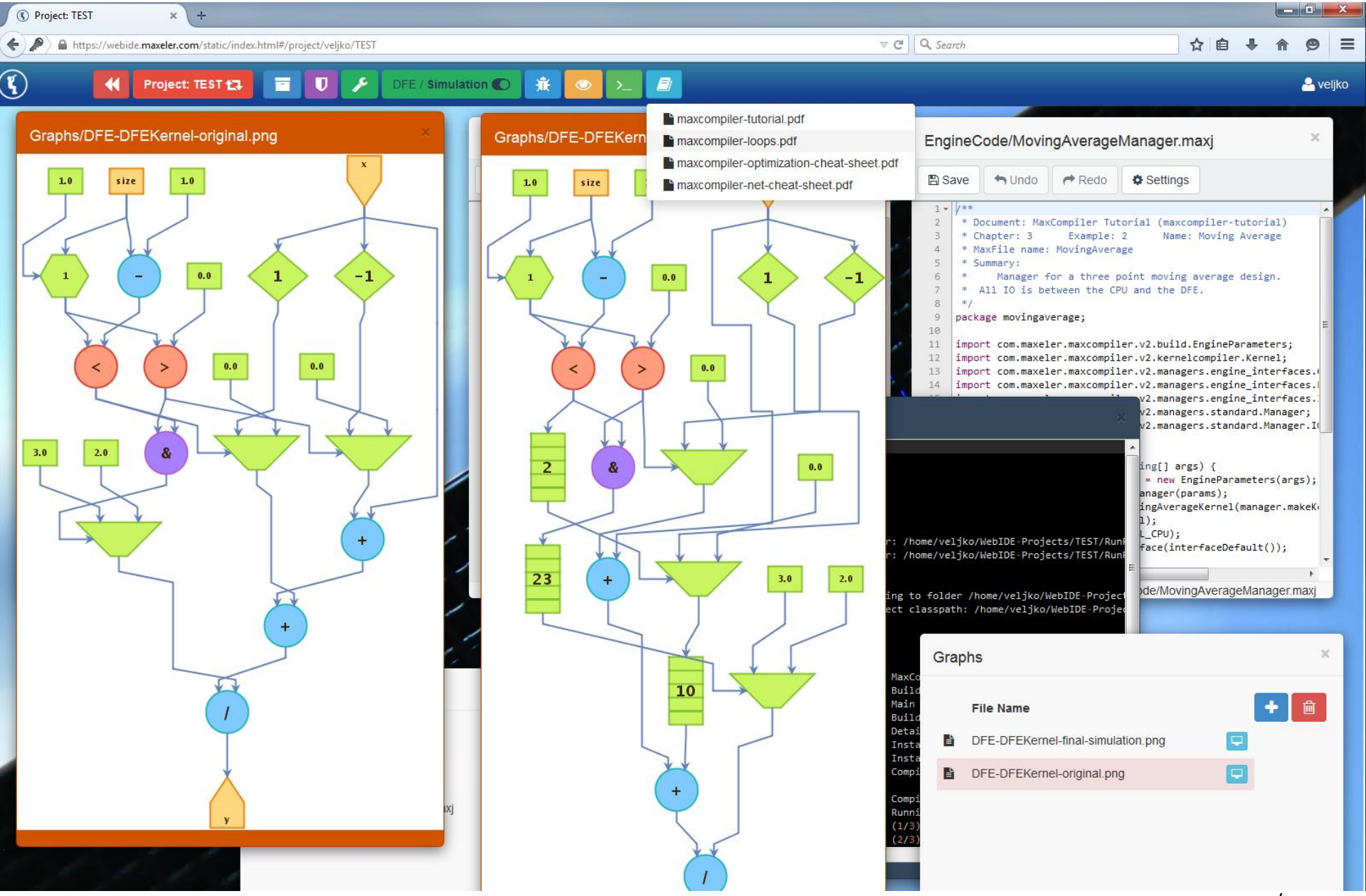

Project Explorer
https://webide.maxeler.com/static/index.html#/projects
Search
veljko
Create new project
TEST
Tutorial: Chapter 03 - Example2 - MovingAverage
-- choose project template --
Default Template
Tutorial: Chapter 03 - Example1 - MovingAverageSimple
Tutorial: Chapter 03 - Example2 - MovingAverage
Tutorial: Chapter 03 - Example3 - MovingAverageWeighted
Tutorial: Chapter 04 - Example1 - PassThrough
Tutorial: Chapter 04 - Example2 - SimpleKernel
Tutorial: Chapter 05 - Example1 - Watches
Tutorial: Chapter 05 - Example2 - Printf
Tutorial: Chapter 06 - Example1 - TypeCast
Tutorial: Chapter 06 - Example2 - Complex
Tutorial: Chapter 06 - Example3 - Vectors
Tutorial: Chapter 07 - Example1 - AddScalar
Tutorial: Chapter 08 - Example1 - SimpleOffset
Tutorial: Chapter 08 - Example2 - TwoDAverageStatic
Tutorial: Chapter 08 - Example3 - TwoDAverageVariable
Tutorial: Chapter 08 - Example4 - ThreeDAverageVariable
Tutorial: Chapter 08 - Example5 - TwoDAverageDynamic
Tutorial: Chapter 08 - Example6 - NormalMoveOut
Tutorial: Chapter 08 - Example7 - StreamHold
MATF
Toronto
MISANU
Maribor


Project: TEST
https://webide.maxeler.com/static/index.html#/project/veljko/TEST
Search
Project: TEST
DFE / Simulation
veljko
Source Code
CPUCode
EngineCode
Welcome
Save
Undo
Redo
Settings
1 ----------
2 Welcome to WebIDE
3 ----------
4
5 INTRODUCTION:
6
7 - Windows
8 * To move window around drag it's top part.
9 * To resize window drag botom right corner.
10 - Folder explorer
11 * Window that shows files inside of the folder.
12 * Double click on file inside file explorer
13 to open that file inside of the editor.
14 * Click on + button to add new file to the folder.
15 * Click on delete buttom to delete file from folder.
16 + HINT: press shift button to select multiple files.
17 - Editor
18 * Window in which you can edit your files.
19 * Closing editor will save your file on the server.
20 * Press CTR+F to use find functionalty.
21 - Image Viewer
22 * Window in which you can view images.
23 * Image viewer window is not resizable.
24 - PDF Viewer
25 * Window in which you can view pdf files.
26
27 GETTING STARTED:
28
uri:Welcome
MAXELER
Technologies


Project: TEST
https://webide.maxeler.com/static/index.html#/project/veljko/TEST
Search
Project: TEST
DFE / Simulation
veljko
CPUCode/cpu_code.c
Save
Undo
Redo
Settings
/**
* Document: MaxCompiler Tutorial (maxcompiler-tutorial)
* Chapter: 3 Example: 2 Name: Moving Average
* MaxFile name: MovingAverage
* Summary:
* CPU code for the three point moving average design.
*/
#include "DFE.h" // Includes .max files
#include <MaxSLiCInterface.h> // Simple Live CPU interface
float dataIn[8] = { 1, 0, 2, 0, 4, 1, 8, 3 };
float dataOut[8];
int main()
{
printf("Running DFE\n");
DFE(8, dataIn, dataOut);
for (int i = 1; i < 7; i++) // Ignore edge values
printf("dataOut[%d] = %f\n", i, dataOut[i]);
return 0;
}
uri:/v0.1/file/veljko/TEST/CPUCode/cpu_code.c
EngineCode/MovingAverageKernel.maxj
Save
Undo
Redo
Settings
/**
* Document: MaxCompiler Tutorial (maxcompiler-tutorial.pdf)
* Chapter: 3
* Example: 2
* Summary:
* Computes a three point moving average using zero at
*/
package movingaverage;
import com.maxeler.maxcompiler.v2.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v2.kernelcompiler.KernelParame
import com.maxeler.maxcompiler.v2.kernelcompiler.types.base.D
class MovingAverageKernel extends Kernel {
MovingAverageKernel(KernelParameters parameters) {
super(parameters);
// Input
DFEVar x = io.input("x", dfeFloat(8, 24));
DFEVar size = io.scalarInput("size", dfeUInt(32));
// Data
DFEVar prevOriginal = stream.offset(x, -1);
DFEVar nextOriginal = stream.offset(x, 1);
uri:/v0.1/file/veljko/TEST/EngineCode/MovingAverageKernel.maxj
EngineCode/MovingAverageManager.maxj
Save
Undo
Redo
Settings
/**
* Document: MaxCompiler Tutorial (maxcompiler-tutorial)
* Chapter: 3 Example: 2 Name: Moving Average
* MaxFile name: MovingAverage
* Summary:
* Manager for a three point moving average design.
* All IO is between the CPU and the DFE.
*/
package movingaverage;
import com.maxeler.maxcompiler.v2.build.EngineParameters;
import com.maxeler.maxcompiler.v2.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v2.managers.standard.Manager;
class MovingAverageManager {
public static void main(String[] args) {
EngineParameters params = new EngineParameters(args);
Manager manager = new Manager(params);
manager.setKernel(kernel);
manager.setIO(IOType.ALL_CPU);
manager.createSLiCinterface(interfaceDefault());
manager.build();
uri:/v0.1/file/veljko/TEST/EngineCode/MovingAverageManager.maxj
EngineCode
File Name
MovingAverageKernel.maxj
MovingAverageManager.maxj

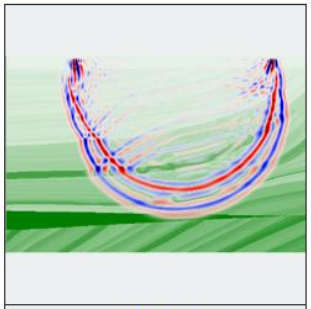

Project: TEST
https://webide.maxeler.com/static/index.html#/project/veljko/TEST
Search
Project: TEST
DFE / Simulation
veljko
BUILD
RUN
CPUCode/cpu_code.c
Save
Undo
Redo
Settings
/**
* Document: MaxCompiler Tutorial (maxcompiler-tutorial)
* Chapter: 3 Example: 2 Name: Moving Average
* MaxFile name: MovingAverage
* Summary:
* CPU code for the three point moving average design.
*/
#include "DFE.h" // Includes .max files
#include <MaxSLiCInterface.h> // Simple Live CPU interface
float dataIn[8] = { 1, 0, 2, 0, 4, 1, 8, 3 };
float dataOut[8];
int main()
{
printf("Running DFE\n");
DFE(8, dataIn, dataOut);
for (int i = 1; i < 7; i++) // Ignore edge values
printf("dataOut[%d] = %f\n", i, dataOut[i]);
return 0;
}
uri:/v0.1/file/veljko/TEST/CPUCode/cpu_code.c
EngineCode/MovingAverageKernel.maxj
Save
Undo
Redo
Settings
/**
* Document: MaxCompiler Tutorial (maxcompiler-tutorial.pdf)
* Chapter: 3
* Example: 2
* Summary:
* Computes a three point moving average using zero at
*/
package movingaverage;
import com.maxeler.maxcompiler.v2.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v2.kernelcompiler.KernelParame
import com.maxeler.maxcompiler.v2.kernelcompiler.types.base.D
class MovingAverageKernel extends Kernel {
MovingAverageKernel(KernelPara
super(parameters);
// Input
DFEVar x = io.input("x", d
DFEVar size = io.scalarInp
// Data
DFEVar prevOriginal = strea
DFEVar nextOriginal = strea
uri:/v0.1/file/veljko/TEST/EngineCo
EngineCode/MovingAverageManager.maxj
Save
Undo
Redo
Settings
/**
* Document: MaxCompiler Tutorial (maxcompiler-tutorial)
* Chapter: 3 Example: 2 Name: Moving Average
* MaxFile name: MovingAverage
* Summary:
* Manager for a three point moving average design.
* All IO is between the CPU and the DFE.
*/
package movingaverage;
import com.maxeler.maxcompiler.v2.build.EngineParameters;
import com.maxeler.maxcompiler.v2.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v2.managers.engine_interfaces.
import com.maxeler.maxcompiler.v2.managers.engine_interfaces.
v2.managers.standard.Manager;
v2.managers.standard.Manager.I
ing[] args) {
= new EngineParameters(args);
anager(params);
ingAverageKernel(manager.makeK
1);
L_CPU);
face(interfaceDefault());
ode/MovingAverageManager.maxj
Output
Buildfile: max.xml
build:
clean:
build:
[mkdir] Created dir: /home/veljko/WebIDE-Projects/TEST/Run
[mkdir] Created dir: /home/veljko/WebIDE-Projects/TEST/Run
[maxjcompiler]
[maxjcompiler]
[maxjcompiler] Compiling to folder /home/veljko/WebIDE-Projec
[maxjcompiler] incorrect classpath: /home/veljko/WebIDE-Proje
all:
run:
[java] Tue 13:18: MaxCompiler version: 2013.3
[java] Tue 13:18: Build "DFE" start time: Tue Apr 28 13:1
[java] Tue 13:18: Main build process running as user root
[java] Tue 13:18: Build location: /home/veljko/WebIDE-Bui
[java] Tue 13:18: Detailed build log available in "_build
[java] Tue 13:18: Instantiating manager
[java] Tue 13:18: Instantiating kernel "DFEKernel"
[java] Tue 13:18: Compiling manager (CPU I/O Only)
[java] Tue 13:18:
[java] Tue 13:18: Compiling kernel "DFEKernel"
[java] Tue 13:18: Running back-end simulation build (3 pha
[java] Tue 13:18: (1/3) - Prepare MaxFile Data (GenerateMa
[java] Tue 13:18: (2/3) - Compile Simulation Modules (Sim
uri:Output
EngineCode
File Name
MovingAverageKernel.maxj
MovingAverageManager.maxj


Project: TEST
https://webide.maxeler.com/static/index.html#/project/veljko/TEST
Search
Project: TEST
DFE / Simulation
veljko
Graphs/DFE-DFEKernel-original.png
Graphs/DFE-DFEKernel-final-simulation.png
EngineCode/MovingAverageManager.maxj
EngineCode/MovingAverageManager.maxj
Settings
Graphs
File Name
DFE-DFEKernel-final-simulation.png
DFE-DFEKernel-original.png


Project: TEST
https://webide.maxeler.com/static/index.html#/project/veljko/TEST
Search
Project: TEST
DFE / Simulation
veljko
Graphs/DFE-DFEKernel-original.png
Graphs/DFE-DFEKern
maxcompiler-tutorial.pdf
maxcompiler-loops.pdf
maxcompiler-optimization-cheat-sheet.pdf
maxcompiler-net-cheat-sheet.pdf
EngineCode/MovingAverageManager.maxj
Save
Undo
Redo
Settings
/**
* Document: MaxCompiler Tutorial (maxcompiler-tutorial)
* Chapter: 3 Example: 2 Name: Moving Average
* MaxFile name: MovingAverage
* Summary:
* Manager for a three point moving average design.
* All IO is between the CPU and the DFE.
*/
package movingaverage;
import com.maxeler.maxcompiler.v2.build.EngineParameters;
import com.maxeler.maxcompiler.v2.kernelcompiler.Kernel;
import com.maxeler.maxcompiler.v2.managers.engine_interfaces.
import com.maxeler.maxcompiler.v2.managers.engine_interfaces.
v2.managers.engine_interfaces.
v2.managers.standard.Manager;
v2.managers.standard.Manager.I
ing[] args) {
= new EngineParameters(args);
anager(params);
ingAverageKernel(manager.makeK
1);
L_CPU);
face(interfaceDefault());
ode/MovingAverageManager.maxj
Graphs
File Name
DFE-DFEKernel-final-simulation.png
DFE-DFEKernel-original.png

appgallery.maxeler.com/#/

appgallery



3D FD Modeling

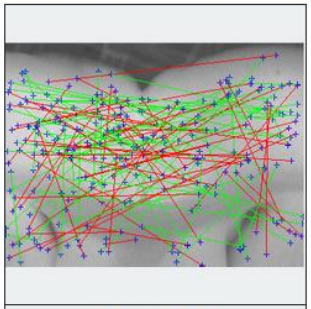
3D finite difference wave modeling.

Author: Maxeler London

Bitcoin Miner

Bitcoin's proof-of-work is implemented by incrementing a nonce in a transaction block until the block header's has

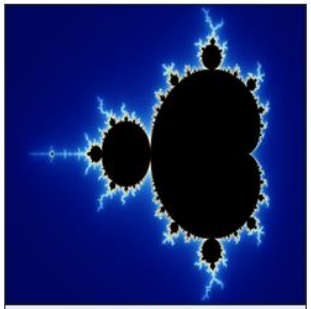
Author: Maxeler London



Brain Network

Linear correlation analysis of brain images to detect brain activity.

Author: Maxeler London



Fractal

Generate the Mandelbrot and Julia sets.

Author: Maxeler London

Jacobi Solver

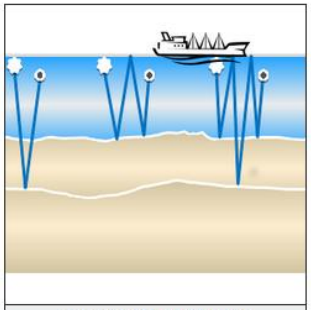
The Jacobi App implements a solver for equations of the type Ax=b, where A is constant but where we have a set

Author: Maxeler London

N-Body Simulation

The N-Body App simulates interactions between N particles under gravitational forces in space. A particle's state

Author: Maxeler London



Reverse Time Migration

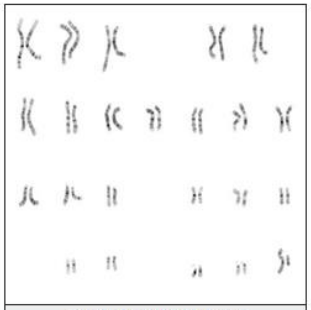
Real time seismic monitoring of hydraulic fracturing sites, more efficient subsurface exploration and precisio

Author: Maxeler London

Single Step Monte-Carlo

Compute the expectation from a Monte Carlo simulation sampling over a basket of items that can be modelled through

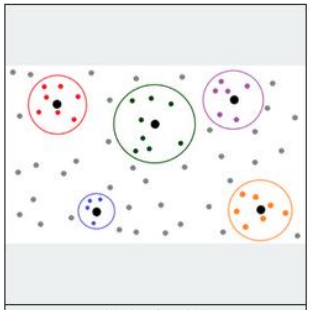
Author: Maxeler London



Smith Waterman Demo

Smith Waterman is a standard textbook algorithm for local gene sequence alignment. While it is impractical fo

Author: Maxeler London



Classification

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called

Author: Maxeler Networking

appgallery.maxeler.com

http://www.mi.sanu.ac.rs/~appgallery.maxeler/

How to Prepare an Application for AppGallery?

Call for appgallery.maxeler.com
Open to all!

An AppGallery application
should include the following:

- Photo of the fresco
- Title for the fresco
- Description (the first few words go to the fresco)
- PDF TECH (technical documentation)
- PDF USER (use case documentation)
- GitHub app binaries for a Maxeler system (one or more)
- Video or animation (optional)
- CV of the author

The open-source applications
should be placed on GitHub.

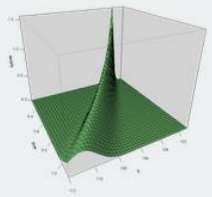
More on how to prepare code for GitHub:
https://github.com/maxeler/MaxAppTemplate

Apps | Maxeler AppGallery

appgallery.maxeler.com/#/

Search

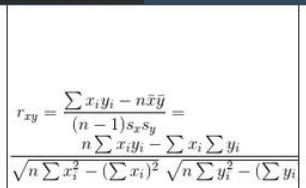
Apps | Filter by: | Maximum Performance Eco-System, Beta Release (v0.1) | Log In



Black-Scholes Option Pricer

This App uses a Monte-Carlo simulation for tail risk analysis. Given a horizon time, a set of up to 6 underliers and

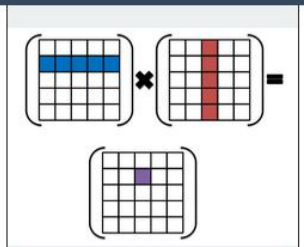
Author: Maxeler Analytics



Correlation

Correlation is a statistical measure that indicates the extent to which two or more variables fluctuate together.

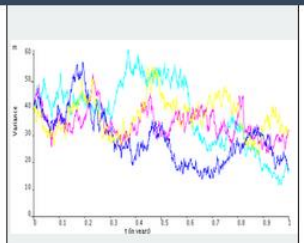
Author: Maxeler Analytics



Dense Matrix Multiplication

A matrix is dense if most of its elements are non-zero. Multiplication involves a dot product between every row of

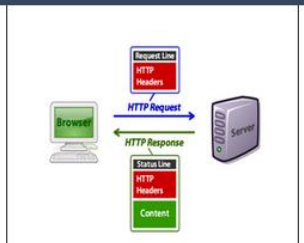
Author: Maxeler Analytics



Heston option pricer

Monte Carlo options pricer.

Author: Maxeler Analytics



Low-Latency HTTP Web-Server

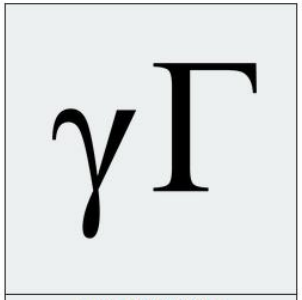
This App implements an HTTP Web-Server in a DFE. The App serves static webpages directly from LMEM

Author: Maxeler Belgrade

Non-Central Chi Square

Non-Central Chi Square Distribution, that generate lots of sample from complex distribution.

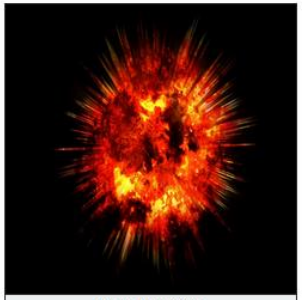
Author: Maxeler California



Gamma Distribution

Generate random numbers according to the Gamma statistical distribution, using the Gamma rejection method. As

Author: Maxeler Intern



Heat Equation

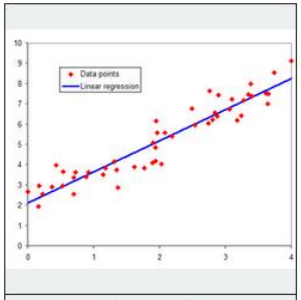
Simulate heat spreading according to explicit methods, which is characterized by the following equation: T(p, t+1) = dt

Author: Maxeler Intern

Implicit Heat Equation

Simulate heat spreading according to implicit methods, which is characterized by the following equations: b = T(p, t)

Author: Maxeler Intern



Linear Regression

In statistics, linear regression is an approach for modeling the relationship between a scalar dependent variable and

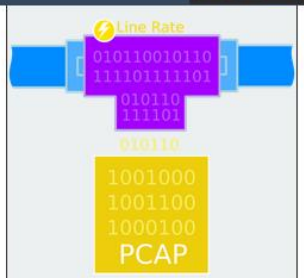
Author: Maxeler Intern

Apps | Maxeler AppGallery

appgallery.maxeler.com/#/

Search

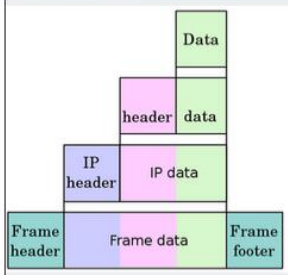
Apps | Filter by: | Maximum Performance Eco-System, Beta Release (v0.1) | Log In



High Speed Packet Capture

Provides line-rate packet capture at bursts of up to 24GB in size. The application configures pairs of DFE SFI

Author: Maxeler Networking



Regex

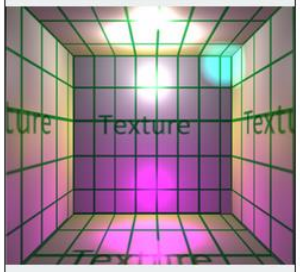
The Regex app takes a regular expression and builds a state machine to implement the regular expression

Author: Maxeler Networking

JPEG Decoder

The JPEG compression algorithm is at its best on photographs and paintings of realistic scenes with smooth variation

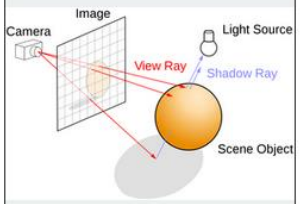
Author: Marko Stupar



Ray Casting

In computer graphics, ray tracing is a technique for generating an image by tracing the path of light through pixels

Author: Marko Stupar

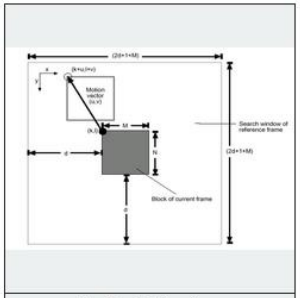


Click to see more details!

Ray Tracing

In computer graphics, ray tracing is a technique for generating an image by tracing the path of light through pixels

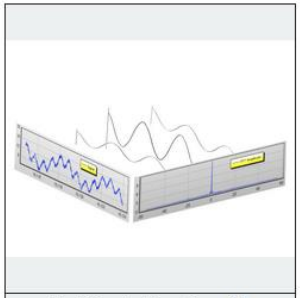
Author: Marko Stupar



Motion Estimation

Motion Estimation is used in video encoding to describe a video frame by motion vectors from other frames

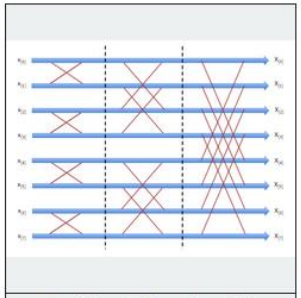
Author: Maxeler Intern



Fast Fourier Transform 2D

This application performs a 2D Fast Fourier Transform on a two dimensional array of complex numbers.

Author: Maxeler London



Fast Fourier Transform 1D

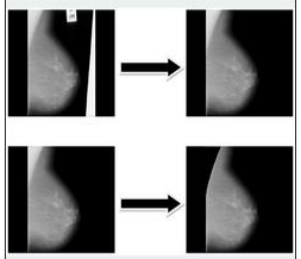
This application performs a one dimensional Fast Fourier Transform on a one dimensional array of complex

Author: Maxeler London

Hybrid Coin Miner

A coin miner application that can mine SHA-256 based coins and Scrypt based coins simultaneously.

Author: Maxeler London



Breast Mammogram ROI Extraction

This App extracts the region of interest from breast mammogram images. Basically, this app removes pectoral

Author: Faculty of Engineering, University of Kragujevac, BioIRC Ltd Kragujevac

Apps | Maxeler AppGallery | imilankovic/Breast-Mamm...

GitHub, Inc. (US) https://github.com/imilankovic/Breast-Mammogram-ROI-Extraction

Search

GitHub This repository Search Explore Features Enterprise Blog Sign up Sign in

imilankovic / Breast-Mammogram-ROI-Extraction Watch 1 Star 1 Fork 0

Breast Mammogram ROI Extraction App

4 commits 1 branch 0 releases 1 contributor

branch: master Breast-Mammogram-ROI-Extraction / +

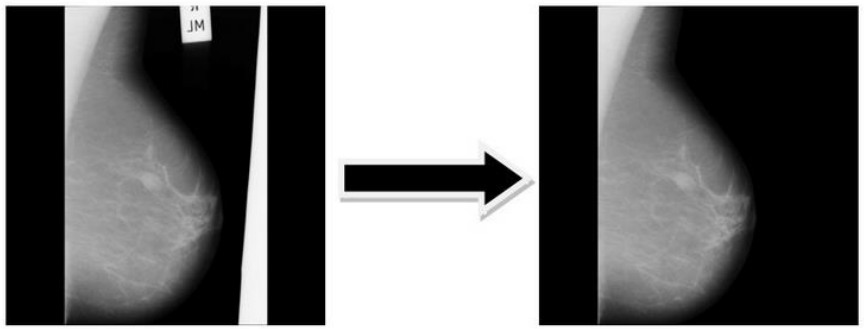
Updated README.

imilankovic authored 23 hours ago latest commit 35c69d1e30

APP	First Commit	a day ago
DOC	Updated DOC.	23 hours ago
README.md	Updated README.	23 hours ago

README.md

Breast Mammogram ROI Extraction

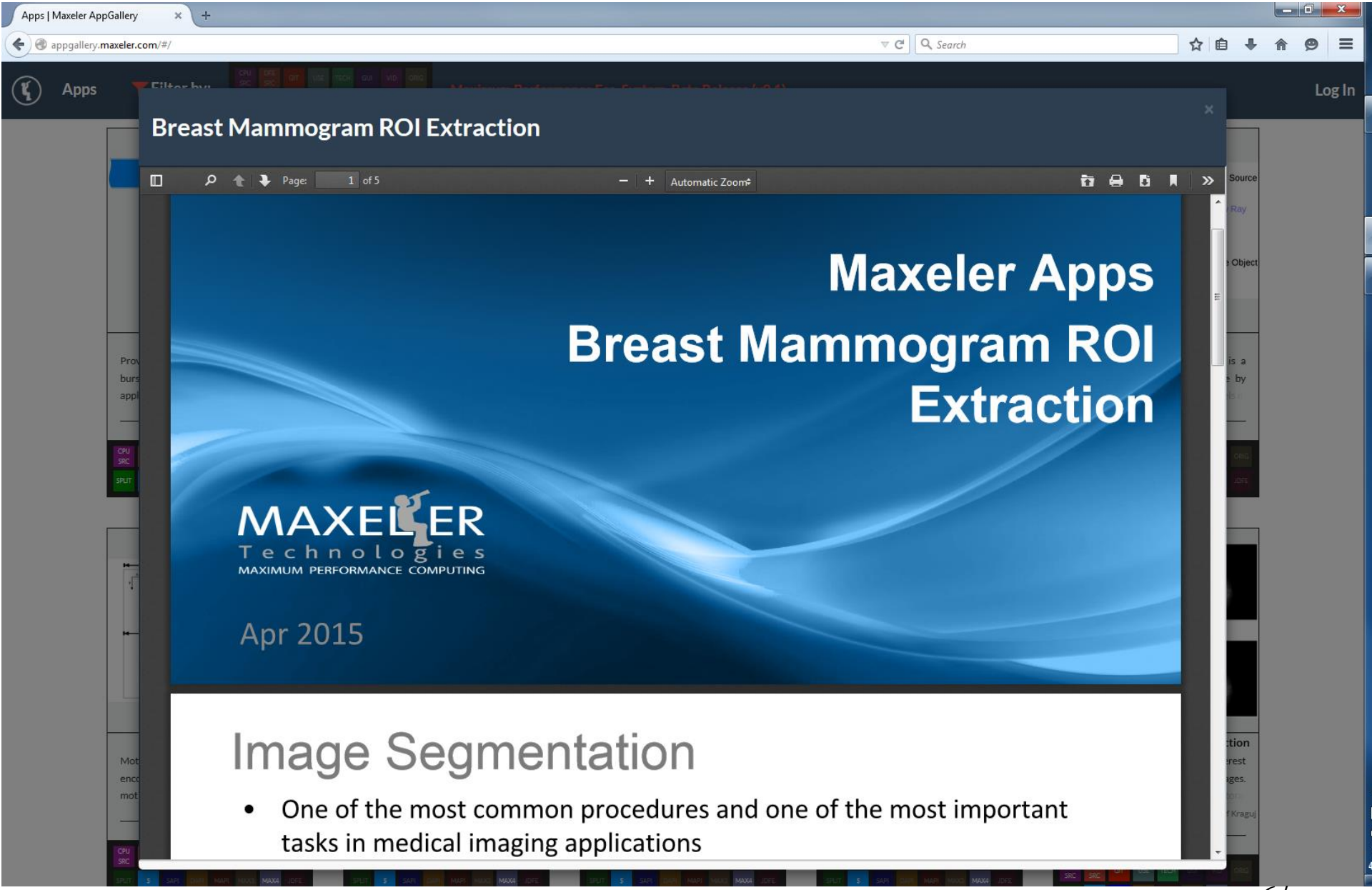


Code
Issues 0
Pull requests 0
Pulse
Graphs

HTTPS clone URL
https://github.com/i

You can clone with HTTPS or Subversion.

Clone in Desktop
Download ZIP


Apps | Maxeler AppGallery
appgallery.maxeler.com/#/
Search
Apps
Log In
Breast Mammogram ROI Extraction
Page: 1 of 5
Automatic Zoom
Maxeler Apps
Breast Mammogram ROI Extraction
MAXELER Technologies
MAXIMUM PERFORMANCE COMPUTING
Apr 2015
Image Segmentation
• One of the most common procedures and one of the most important tasks in medical imaging applications

Apps | Maxeler AppGallery

appgallery.maxeler.com/#/

Search

Apps

Log In

Breast Mammogram ROI Extraction

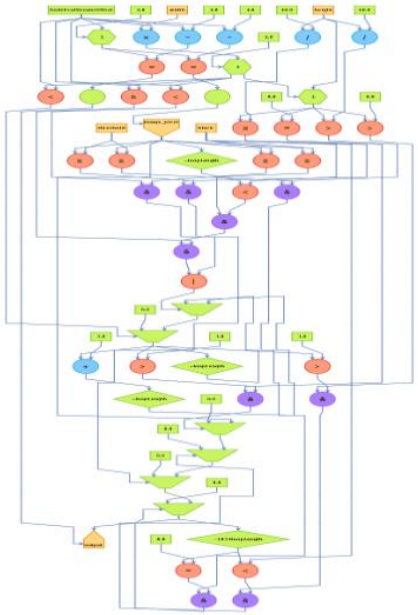
Page: 9 of 12

Automatic Zoom

8

MAXELER Technologies

Kernel Graph



9

MAXELER Technologies

Apps | Maxeler AppGallery

appgallery.maxeler.com/#/

Search

Apps

Log In

Breast Mammogram ROI Extraction

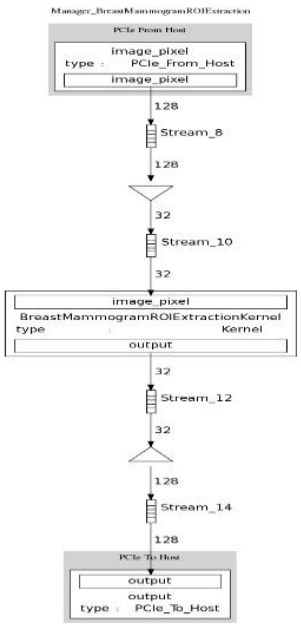
Page: 10 of 12

Automatic Zoom

9

MAXELER Technologies

Manager Graph



10

MAXELER Technologies

MaxGenFD: Isotropic Modeling

$$\frac{\partial^2 p}{\partial t^2} = K \vec{\nabla} \cdot \left(\frac{1}{\rho} \vec{\nabla} p \right) + S(t)$$

FDKernel (.java)

```
public class IsotropicModelingKernel extends FDKernel {
    public IsotropicModelingKernel(FDKernelParameters p) {
        super(p);
        Stencil stencil = fixedStencil(−6, 6, coeffs, 1/8.0);
        FDVar curr = io.wavefieldInput("curr", 1.0, 6);
        FDVar prev = io.wavefieldInput("prev", 1.0, 0);
        FDVar dvv = io.earthModelInput("dvv", 9.0, 0);
        FDVar source = io.hostInput("source", 1.0, 0);

        FDVar l = convolve(curr, ConvolveAxes.XYZ, stencil);

        FDVar sponge = boundaries.sponge(50);
        prev = prev * sponge;
        FDVar next = curr * 2 − prev + dvv * l + source;
        next = next * sponge;

        io.wavefieldOutput("next", next);
    }
}
```

Host Code (.c)

```
for(t = 1; t < tmax; t++) {
    // Set−up timestep
    if (t < tsrc) {
        source = generate_source_wavelet(t);
        maxlib_stream_region_from_host(maxlib, "source", source
            srcx, srcy, srcz, srcx+1, srcy+1, srcz+1);
    }
    maxlib_stream_from_dram(maxlib, "curr", curr_ptr);
    maxlib_stream_from_dram(maxlib, "prev", prev_ptr);
    maxlib_stream_earthmodel_from_dram(maxlib, dvv_array);

    maxlib_stream_to_dram(maxlib, "next", next_ptr);

    maxlib_run(maxlib); // Execute timestep

    swap_buffers(prev_ptr, curr_ptr, next_ptr);
}
```

Two DataFlow Stories

- The Story of Google and MapReduce
- The Story of Intel and Altera

WHAT IS THE MORALE OF BOTH STORIES?

NEWS

6/27/2014
10:09 AM

Google I/O: Hello Dataflow, Goodbye MapReduce

Google introduces Dataflow to handle streams and batches of big data, replacing MapReduce and challenging other public cloud services.

Charles Babcock
News

Google I/O this year was overwhelmingly dominated by consumer technology, the end user interface, and extension of the Android universe into a new class of mobile devices, the computer you wear on your wrist.

Hadoop Jobs: 9 Ways To Get Hired

(Click image for larger view and slideshow.)

At the same time, there were one or two enterprise-scale data handling and cloud computing gems scattered among all the end user announcements.

Intel drops talks to acquire Altera: Sources

Everett Rosenfeld | @Ev_Rosenfeld

Thursday, 9 Apr 2015 | 9:06 AM ET

CNBC

Getty Images

Visitors take pictures next to the Intel logo at Intel headquarters in Santa Clara, Calif.

Intel has stopped its talks with Altera about a potential acquisition, according to sources familiar with the situation.

Altera stock fell about 10 percent in pre-market trading. When news of the talks first broke in March, Altera stock posted a 28 percent jump.

The two companies have not spoken in more than a week, sources said, as neither party could agree on a price.

Read More › Altera shares jumped 28% on Intel interest

Some Useful Links on DataFlow R+D

ORIGINAL BOOK ON DATAFLOW RESEARCH:

Amazon:
http://www.amazon.com/Guide-DataFlow-Supercomputing-Concepts-Communications/dp/3319162284

Springer:
http://www.springer.com/gp/book/9783319162287

EDITED BOOK ON DATAFLOW CASE STUDIES:

Amazon:
http://www.amazon.com/Dataflow-Processing-Volume-Advances-Computers/dp/0128021349

Elsevier:
http://www.elsevier.com/books/dataflow-processing/milutinovic/978-0-12-802134-7

TOOLS AND EXAMPLES SUPPORTING BOTH BOOKS ABOVE:

webide.maxeler.com (id/passw: veljko1951)

appgallery.maxeler.com

ABOUT GOOGLE DROPPING MAPREDUCE AND INTEL TRYING ALTERA:

http://www.informationweek.com/cloud/software-as-a-service/google-i-o-hello-dataflow-goodbye-mapreduce/d/d-id/1278917

http://www.cnbc.com/id/102546244

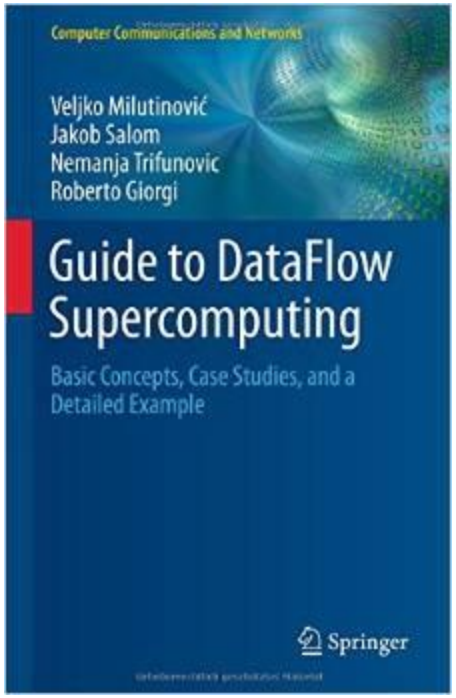
TWO COAUTHORED PAPERS:

The IET 2014 Premium Award for Computing and Digital Techniques:
http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?arnumber=6337379

Journal of Big Data:
http://link.springer.com/article/10.1186/s40537-014-0010-z

An Original Book Covering the Essence

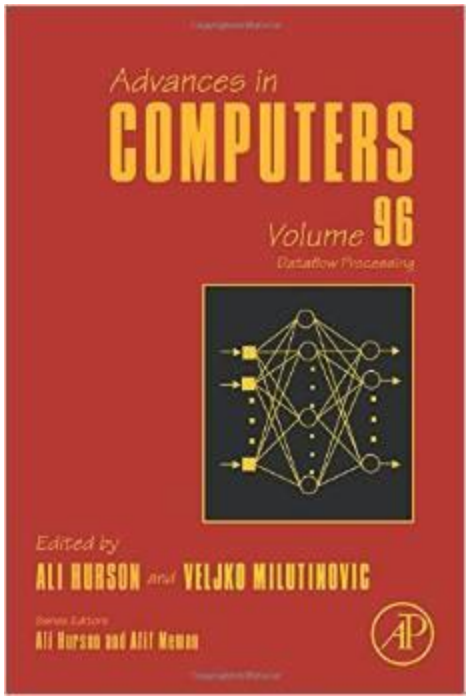
- http://www.amazon.com/Guide-DataFlow-Supercomputing-Concepts-Communications/dp/3319162284
- http://www.springer.com/gp/book/9783319162287



The first source to use the term the Feynman Paradigm in contrast with the Von Neumann Paradigm

An Edited Book Covering the Applications

- http://www.amazon.com/Dataflow-Processing-Volume-Advances-Computers/dp/0128021349
- http://www.elsevier.com/books/dataflow-processing/milutinovic/978-0-12-802134-7



Contributions welcome for the follow-ups: Vol. 102 + Vol. 104

QoL

Maxeler is one of the Top 10 HPC projects
to impact QoL in the World :)

Scientific Computing
[www.scientificcomputing.com/articles/2014/11]

by

Don Johnson

of

Lawrence Livermore National Labs
[editor@ScientificComputing.com]

MindGenomics by Howard Moskowitz

Malcolm Gladwell TED Ideas Worth Spreading (Total Views by 2015: 4,505,218)

Identify topic area (BigData + LowPower)

Raw Materials – Silos & Elements

INTERNET

IdeaMap™

SURVEY

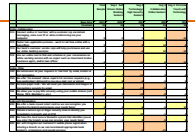
Target Customers

- Impact on Customer Experience

ANALYZED SURVEY RESULTS → Mind Genomics (Maxeler)

MARKETING PHRASES

MARKET SEGMENTATION



TYPING ENGINE

Q&A

SRB=#1@THE(Q/GNP):
http://www.timeshighereducation.co.uk/news/uk-is-a-knockout-performer-on-pound-for-pound-basis/2020319.article
About Times Higher Education:
http://en.wikipedia.org/wiki/Times_Higher_Education

vm@etf.rs

USNewsReport, MATH(CS): #366, #87, #29, #5